

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 2月19日

出 願 番 号

Application Number:

特願2003-041838

[ ST.10/C ]:

[ JP 2003-041838 ]

出 願 人

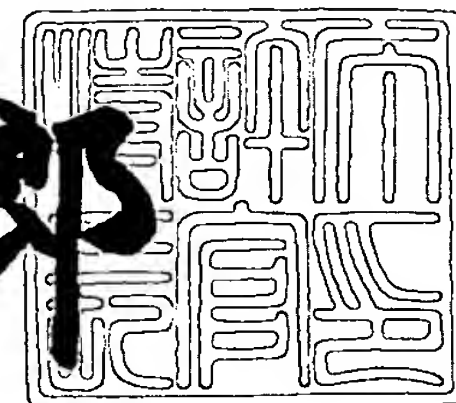
Applicant(s):

日本電気株式会社

2003年 6月20日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3048672

【書類名】 特許願

【整理番号】 33509975

【あて先】 特許庁長官殿

【特記事項】 特許法第 3 0 条第 1 項の規定の適用を受けようとする特  
許出願

【国際特許分類】 H04L 12/00  
H04M 7/00

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 榎本 敦之

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 厩橋 正樹

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 飛鷹 洋一

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 岩田 淳

【発明者】

【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

【氏名】 渋谷 真

【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

【代理人】

【識別番号】 100093595

【弁理士】

【氏名又は名称】 松本 正夫

【先の出願に基づく優先権主張】

【出願番号】 特願2002-242621

【出願日】 平成14年 8月22日

【手数料の表示】

【予納台帳番号】 057794

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9303563

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラム

【特許請求の範囲】

【請求項 1】 複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、

ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とするノード。

【請求項 2】 前記ネットワークの構成変更が、ノードの追加又は削除やリンクトポロジの変化であることを特徴とする請求項 1 に記載のノード。

【請求項 3】 複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、

ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とするノード。

【請求項 4】 複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、

独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、

転送に利用しているスパニングツリーに対応したタグを返答するタグテーブルと、

前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、

転送に利用するスパニングツリーを決定するツリーセクタと、

宛先ごとのフレームの転送出力先を記録するフォワーディングテーブルと、

前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、



前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器とを備えることを特徴とするノード。

【請求項 5】 前記ツリーセレクタが、  
転送に利用するスパニングツリーの切り替えを行うメインコントローラと、  
スパニングツリーの安定を示す指定した時間のタイマ満了を知らせる安定タイマと、  
フレームに付加されたタグを削除するタグ削除器と、  
スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信器と、  
フレームにタグを付加するタグ挿入器とを備えることを特徴とする請求項 4 に記載のノード。

【請求項 6】 前記ツリーセレクタは、  
スパニングツリーの安定を示すフレームの到着間隔を判別するため、一定時間が経過するとタイマ満了通知を送信する到着間隔タイマを備えることを特徴とする請求項 5 に記載のノード。

【請求項 7】 前記ツリーセレクタは、  
リンクコストの算出に利用する指定時間のタイマ満了を知らせるコスト参照タイマを備えることを特徴とする請求項 4 から請求項 6 の何れか 1 項に記載のノード。

【請求項 8】 前記ツリーマネージャが、  
フレームに付加されたタグを削除するタグ削除器と、  
B P D U を送受信する B P D U 送受信器と、  
フレームにタグを付加するタグ挿入器と、  
スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコントローラと、  
前記スパニングツリープロトコルで使用するパラメータを保持するツリーテーブルとを備えることを特徴とする請求項 4 から請求項 7 の何れか 1 項に記載のノード。

【請求項 9】 前記ツリーマネージャは、

通知されたリンクコストに所定の設定値を加算して返答するコスト操作器を備えることを特徴とする請求項 8 に記載のノード。

【請求項 1 0】 リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタを備えることを特徴とする請求項 4 から請求項 9 の何れか 1 項に記載のノード。

【請求項 1 1】 前記リンクコストを、利用状況に基づいて計算することを特徴とする請求項 3 に記載のノード。

【請求項 1 2】 前記利用状況を、空き帯域と定義することを特徴とする請求項 1 1 に記載のノード。

【請求項 1 3】 前記利用状況を、CPU 負荷と定義することを特徴とする請求項 1 1 に記載のノード。

【請求項 1 4】 複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、

ネットワーク内の各ノードがルートノードとなるスパニングツリーを生成し、宛先がルートノードとなるスパニングツリーを用いてフレームを転送することを特徴とするノード。

【請求項 1 5】 複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、

独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、

転送に利用しているスパニングツリーに対応したタグを返答するタグテーブルと、

前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、ネットワークに存在するルートノード数と同数のツリーマネージャを生成するツリーセクタと、

宛先ごとのフレームの転送出力先を記録するフォワーディングテーブルと、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、

前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器

とを備えることを特徴とするノード。

【請求項 1 6】 前記ツリーセクタが、  
ツリーマネージャの作成もしくは削除を行うメインコントローラと、  
フレームに付加されたタグを削除するタグ削除器と、  
スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信器と、

フレームにタグを付加するタグ挿入器とを備えることを特徴とする請求項 1 5  
に記載のノード。

【請求項 1 7】 前記ツリーマネージャが、  
フレームに付加されたタグを削除するタグ削除器と、  
B P D U を送受信する B P D U 送受信器と、  
フレームにタグを付加するタグ挿入器と、  
スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコン  
トローラと、

前記スパニングツリープロトコルで使用するパラメータを保持するツリーテー  
ブルとを備えることを特徴とする請求項 1 5 または請求項 1 6 に記載のノード。

【請求項 1 8】 リンクの接続状況や空き帯域を含むリソース情報を計測す  
るリソースモニタを備えることを特徴とする請求項 1 5 から請求項 1 7 の何れか  
1 項に記載のノード。

【請求項 1 9】 複数のノードを接続したネットワーク上でスパニングツリ  
ーを構成するノードにおいて、

スパニングツリーを生成するツリーマネージャが、  
スパニングツリープロトコルの種類およびバージョンにより、コスト値を調整  
するコスト操作器を備えることを特徴とするノード。

【請求項 2 0】 前記コスト操作器は、  
障害回復処理が遅いプロトコルを利用しているリンクに対して、大きなコスト  
を割り振ることを特徴とする請求項 1 9 に記載のノード。

【請求項 2 1】 複数のノードを接続したネットワーク上でスパニングツリ  
ーを構成するノードにおいて、

ネットワーク内の動作の遅いプロトコルを使用中のリンクのそれぞれについて、前記各リンクのコストが最大となるスパニングツリーを生成し、前記各リンクに障害が発生した際に、前記リンクのコストが最大となっているツリーを利用してフレーム転送を行うことを特徴とするノード。

【請求項 2 2】 複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、

独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、

転送用ツリーに対応したタグを返答するタグテーブルと、  
前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、  
ネットワークに存在する動作の遅いプロトコルを使用中のリンク数と同数のツリーマネージャを生成するツリーセクタと、

宛先ごとのフレーム転送出力先を記録するフォワーディングテーブルと、  
前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、

前記タグに従って転送先のツリーマネージャを決定する分別器  
とを備えることを特徴とするノード。

【請求項 2 3】 前記ツリーセクタが、  
ツリーマネージャの作成もしくは削除を行うツリーセクタ内のメインコントローラと、

フレームに付加されたタグを削除するタグ削除器と、  
制御フレームを送信する G V R P 送受信器と、  
フレームにタグを付加するタグ挿入器とを備えることを特徴とする請求項 2 2 に記載のノード。

【請求項 2 4】 前記ツリーマネージャが、  
フレームに付加されたタグを削除するタグ削除器と、  
B P D U を送受信する B P D U 送受信器と、  
フレームにタグを付加するタグ挿入器と、  
スパニングツリープロトコルにしたがってスパニングツリーを作成するツリー

コントローラと、

スパニングツリープロトコルで使用するパラメータを保持するツリーテーブルとを備えることを特徴とする請求項 2 2 または請求項 2 3 に記載のノード。

【請求項 2 5】 リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタを備えることを特徴とする請求項 2 2 から請求項 2 4 の何れか 1 項に記載のノード。

【請求項 2 6】 スパニングツリープロトコルで利用されている H E L L O フレームより、短い間隔で障害検出用フレームを送受信して障害を検出する障害検出器を備えることを特徴とする請求項 4 から請求項 2 5 の何れか 1 項に記載のノード。

【請求項 2 7】 前記フォワーディングテーブルが、ブロードキャスト出力ポートフィールドを有することを特徴とする、請求項 4、1 5、もしくは 2 2 に記載のノード。

【請求項 2 8】 前記フォワーディングテーブルが、予備出力ポートフィールドを有することを特徴とする、請求項 4、1 5、もしくは 2 2 に記載のノード。

【請求項 2 9】 スパニングツリーによって決定されるポート種別を用いて、出力先ポートを決定することを特徴とする、請求項 4、1 5、もしくは 2 2 に記載のノード。

【請求項 3 0】 前記スパニングツリーによって決定されるポート種別が、ルートポート及びディジグネイテッドポートの何れかであることを特徴とする請求項 2 9 に記載のノード。

【請求項 3 1】 複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、

ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替える処理を実行することを特徴とするスパニングツリー構成プログラム。

【請求項 3 2】 前記ネットワークの構成変更が、ノードの追加又は削除や

リンクトポロジの変化であることを特徴とする請求項 3 1 に記載のスパニングツリー構成プログラム。

【請求項 3 3】 複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、

ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替える処理を実行することを特徴とするスパニングツリー構成プログラム。

【請求項 3 4】 複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、

複数のツリーマネージャによる独立して動作する複数のスパニングツリーを生成する処理と、

転送に利用しているスパニングツリーに対応したタグを返答する処理と、

返答のあった前記タグをフレームに挿入するタグ挿入処理と、

転送に利用するツリーを決定するツリーセクタ処理と、

宛先ごとのフレームの転送出力先を記録するフォワーディングテーブル処理と

、  
前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送処理と、

前記タグに従って転送先のツリーマネージャを決定する分別処理

とを実行することを特徴とするスパニングツリー構成プログラム。

【請求項 3 5】 前記ツリーセクタ処理において、

転送に利用するスパニングツリーの切り替えを行うコントローラ処理と、

スパニングツリーの安定を示す指定した時間のタイマ満了を知らせる安定タイマ処理と、

フレームに付加されたタグを削除するタグ削除処理と、

スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信処理と、

フレームにタグを付加するタグ挿入処理



とを実行することを特徴とする請求項 3 4 に記載のスパニングツリー構成プログラム。

【請求項 3 6】 前記ツリーセクタ処理において、  
スパニングツリーの安定を示すフレームの到着間隔を判別するため、一定時間が経過すると、タイマ満了通知を送信する到着間隔タイマ処理を実行することを特徴とする請求項 3 5 に記載のスパニングツリー構成プログラム。

【請求項 3 7】 前記ツリーセクタ処理は、  
リンクコストの算出に利用する指定時間のタイマ満了を知らせるコスト参照タイマ処理を実行することを特徴とする請求項 3 4 から請求項 3 6 に記載のスパニングツリー構成プログラム。

【請求項 3 8】 前記ツリーマネージャ処理が、  
フレームに付加されたタグを削除するタグ削除処理と、  
B P D U を送受信する B P D U 送受信処理と、  
フレームにタグを付加するタグ挿入処理と、  
スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコントローラ処理と、

前記スパニングツリープロトコルで使用するパラメータを保持するツリーテーブル処理とを実行することを特徴とする請求項 3 4 から請求項 3 7 の何れか 1 項に記載のスパニングツリー構成プログラム。

【請求項 3 9】 前記ツリーマネージャ処理は、  
通知されたリンクコストに所定の設定値を加算して返答するコスト操作処理を実行することを特徴とする請求項 3 8 に記載のスパニングツリー構成プログラム。

【請求項 4 0】 リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタ処理を実行することを特徴とする請求項 3 4 から請求項 3 9 の何れか 1 項に記載のスパニングツリー構成プログラム。

【請求項 4 1】 リンクコストを、利用状況により計算する処理を実行することを特徴とする請求項 3 3 に記載のスパニングツリー構成プログラム。

【請求項 4 2】 前記利用状況を、空き帯域と定義することを特徴とする請

求項 4 1 に記載のスパニングツリー構成プログラム。

【請求項 4 3】 前記利用状況を、CPU 負荷と定義することを特徴とする請求項 4 1 に記載のスパニングツリー構成プログラム。

【請求項 4 4】 複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、ネットワーク内の各ノードがルートノードとなるスパニングツリーを生成し、宛先がルートノードとなるツリーを用いてフレームを転送することを特徴とするスパニングツリー構成プログラム。

【請求項 4 5】 複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャ処理と、

転送に利用しているスパニングツリーに対応したタグを返答するタグテーブル処理と、

前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入処理と、

ネットワークに存在するルートノード数と同数のツリーマネージャを生成するツリーセクタ処理と、

宛先ごとのフレーム転送出力先を記録するフォワーディングテーブル処理と、

前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送処理と、

前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別処理

とを備えることを特徴とするスパニングツリー構成プログラム。

【請求項 4 6】 前記ツリーセクタ処理が、ツリーマネージャの作成もしくは削除を行うツリーセクタ内のメインコントローラ処理と、

フレームに付加されたタグを削除するタグ削除処理と、

スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信



処理と、

フレームにタグを付加するタグ挿入処理

とを実行することを特徴とする請求項 4 5 に記載のスパニングツリー構成プログラム。

【請求項 4 7】 前記ツリーマネージャ処理が、

フレームに付加されたタグを削除するタグ削除処理と、

B P D U を送受信する B P D U 送受信処理と、

フレームにタグを付加するタグ挿入処理と、

スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコントローラ処理と、

前記スパニングツリープロトコルで使用するパラメータを保持するツリーテーブル処理とを実行することを特徴とする請求項 4 5 または請求項 4 6 に記載のスパニングツリー構成プログラム。

【請求項 4 8】 各前記ノードは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタ処理を実行することを特徴とする請求項 4 5 から請求項 4 7 の何れか 1 項に記載のスパニングツリー構成プログラム。

【請求項 4 9】 複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノードで動作するスパニングツリー構成プログラムにおいて、ネットワーク内の動作の遅いプロトコルを使用中のリンクのそれぞれについて、前記各リンクのコストが最大となるスパニングツリーを生成し、前記各リンクに障害が発生した際に、前記リンクのコストが最大となっているツリーを利用してフレーム転送を行うことを特徴とするスパニングツリー構成プログラム。

【請求項 5 0】 複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノードで動作するスパニングツリー構成プログラムにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャ処理と、

転送用ツリーに対応したタグを返答するタグテーブル処理と、

前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入処理と

ネットワークに存在する動作の遅いプロトコルを使用中のリンク数と同数のツリーマネージャを生成するツリーセクタ処理と、

宛先ごとのフレームの転送出力先を記録するフォワーディングテーブル処理と、

前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送処理と、

前記タグに従ってフレームの転送先のツリーマネージャを決定する分別処理とを備えることを特徴とするスパニングツリー構成プログラム。

【請求項 5 1】 前記ツリーセクタ処理が、ツリーマネージャの作成もしくは削除を行うツリーセクタ内のメインコントローラ処理と、

フレームに付加されたタグを削除するタグ削除処理と、

制御フレームを送信する G V R P 送受信処理と、

フレームにタグを付加するタグ挿入処理

とを備えることを特徴とする請求項 5 0 に記載のスパニングツリー構成プログラム。

【請求項 5 2】 前記ツリーマネージャ処理が、

フレームに付加されたタグを削除するタグ削除処理と、

B P D U を送受信する B P D U 送受信処理と、

フレームにタグを付加するタグ挿入処理と、

スパニングツリープロトコルにしたがってスパニングツリーを作成するツリーコントローラ処理と、

スパニングツリープロトコルで使用するパラメータを保持するツリーテーブル処理とを備えることを特徴とする請求項 5 0 または請求項 5 1 に記載のスパニングツリー構成プログラム。

【請求項 5 3】 各前記ノードは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタ処理を実行することを特徴とする請求項 5 0 から請求項 5 2 の何れか 1 項に記載のスパニングツリー構成プログラム。

【請求項 5 4】 前記フォワーディングテーブルが、ブロードキャスト出力

ポートフィールドを有することを特徴とする、請求項 3 4、4 5、もしくは 5 0 に記載のスパニングツリー構成プログラム。

【請求項 5 5】 前記フォワーディングテーブルが、予備出力ポートフィールドを有することを特徴とする、請求項 3 4、4 5、もしくは 5 0 に記載のスパニングツリー構成プログラム。

【請求項 5 6】 スパニングツリーによって決定されるポート種別を用いて、出力先ポートを決定することを特徴とする、請求項 3 4、4 5、もしくは 5 0 に記載のスパニングツリー構成プログラム。

【請求項 5 7】 前記スパニングツリーによって決定されるポート種別が、ルートポート及びディジグネイテッドポートの何れかであることを特徴とする請求項 5 6 に記載のスパニングツリー構成プログラム。

【請求項 5 8】 複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、

各前記ノードが、

ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とするネットワークシステム。

【請求項 5 9】 複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、

各前記ノードが、

ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とするネットワークシステム。

【請求項 6 0】 複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、

各前記ノードが、

独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャ

と、

転送に利用しているスパニングツリーに対応したタグを返答するタグテーブル

と、

前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、

転送に利用するスパニングツリーを決定するツリーセクタと、

宛先ごとのフレームの転送出力先を記録するフォワーディングテーブルと、

前記フォワーディングテーブルで指定された転送出力先にフレームを転送する  
フレーム転送器と、

前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器  
とを備えることを特徴とするネットワークシステム。

【請求項 6 1】 リンクコストを、利用状況により計算することを特徴とする  
請求項 5 3 に記載のネットワークシステム。

【請求項 6 2】 複数のノードを接続したネットワーク上でスパニングツリー  
によって転送経路を設定するネットワークシステムにおいて、

ネットワーク内の各ノードがルートノードとなるスパニングツリーを生成し、  
宛先がルートノードとなるツリーを用いてフレームを転送する

ことを特徴とするネットワークシステム。

【請求項 6 3】 複数のノードを接続したネットワーク上でスパニングツリー  
によって転送経路を設定するネットワークシステムにおいて、

独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャ  
と、

転送用ツリーに対応したタグを返答するタグテーブルと、

前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、

ネットワークに存在するノード数と同数のツリーマネージャを生成するツリー  
セクタと、

宛先ごとのフレーム転送出力先を記録するフォワーディングテーブルと、

前記フォワーディングテーブルで指定された転送出力先にフレームを転送する  
フレーム転送器と、

前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器

とを備えることを特徴とするネットワークシステム。

【請求項 6 4】 複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、  
スパニングツリーを生成するツリーマネージャが、  
スパニングツリープロトコルの種類およびバージョンにより、コスト値を調整するコスト操作処理  
を実行することを特徴とするネットワークシステム。

【請求項 6 5】 複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、  
スパニングツリーを生成するツリーマネージャが、  
スパニングツリープロトコルの種類およびバージョンにより、コスト値を調整するコスト操作器  
を備えることを特徴とするネットワークシステム。

【請求項 6 6】 複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、  
ネットワーク内の動作の遅いプロトコルを使用中のリンクのそれぞれについて、前記各リンクのコストが最大となるスパニングツリーを生成し、前記各リンクに障害が発生した際に、前記リンクのコストが最大となっているツリーを利用してフレーム転送を行う  
ことを特徴とするネットワークシステム。

【請求項 6 7】 複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、  
独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、  
転送用ツリーに対応したタグを返答するタグテーブルと、  
前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、  
ネットワークに存在する動作の遅いプロトコルを使用中のリンク数と同数のツリーマネージャを生成するツリーセクタと、  
宛先ごとのフレーム転送出力先を記録するフォワーディングテーブルと、

前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、

前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器とを備えることを特徴とするネットワークシステム。

【請求項 6 8】 前記フォワーディングテーブルが、ブロードキャスト出力ポートフィールドを有することを特徴とする、請求項 6 0、6 3、もしくは 6 7 に記載のネットワークシステム。

【請求項 6 9】 前記フォワーディングテーブルが、予備出力ポートフィールドを有することを特徴とする、請求項 6 0、6 3、もしくは 6 7 に記載のネットワークシステム。

【請求項 7 0】 スパニングツリーによって決定されるポート種別を用いて、出力先ポートを決定することを特徴とする、請求項 6 0、6 3、もしくは 6 7 に記載のネットワークシステム。

【請求項 7 1】 前記スパニングツリーによって決定されるポート種別が、ルートポート及びディジグネイテッドポートの何れかであることを特徴とする請求項 7 0 に記載のネットワークシステム。

【請求項 7 2】 複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、

ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とするスパニングツリー構成方法。

【請求項 7 3】 複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、

ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とするスパニングツリー構成方法。

【請求項 7 4】 複数のノードを接続したネットワークにおけるスパニング



ツリー構成方法であって、

新規ノードを追加する際に、新規ノードを既存スパニングツリーに参加させず、予備スパニングツリーにのみ参加させることを特徴とするスパニングツリー構成方法。

【請求項 7 5】 複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、

ノードを削除する際に、削除ノードを予備スパニングツリーに参加させず、既存スパニングツリーのみに参加させることを特徴とするスパニングツリー構成方法。

【請求項 7 6】 複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、

ネットワーク構成が変更された時、予備系を用いて変更後のツリー作成を行うことを特徴とするスパニングツリー構成方法。

【請求項 7 7】 複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、

コスト算出に、リンクの空き帯域を利用することを特徴とするスパニングツリー構成方法。

【請求項 7 8】 複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、

ネットワーク内の全ノードが、全ノードをメンバに持つスパニングツリーのうち、どれか 1 つのルートノードとなるよう、複数のスパニングツリーを作成することを特徴とするスパニングツリー構成方法。

【請求項 7 9】 複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、

ネットワーク内に存在する全ノードをメンバに持つスパニングツリーを作成し、うち、障害復旧の遅いプロトコルを使用しているリンクごとに複数のスパニングツリーを作成することを特徴とするスパニングツリー構成方法。

【請求項 8 0】 複数のノードを接続したネットワークにおいて信号伝送に用いる論理トポロジを形成する方法であって、

ネットワークの構成変更前の論理トポロジを用いて信号伝送を行っている状態で、ネットワークの構成変更後の論理トポロジを生成し、

前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替えることを特徴とする論理トポロジ形成方法。

【請求項 8 1】 自らが属するネットワークにおいて信号伝送に用いる論理トポロジを用いて信号伝送を行っている状態で、前記ネットワークの構成変更時に、ネットワークの構成変更後の論理トポロジを生成する手段と、

前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替える手段とを有するノード。

【請求項 8 2】 自らが属するネットワークにおいて信号伝送に用いる論理トポロジを用いて信号伝送を行っている状態で、前記ネットワークの構成変更時に、ネットワークの構成変更後の論理トポロジを生成する処理と、

前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替える処理とを実行することを特徴とするプログラム。

【請求項 8 3】 複数のノードを接続したネットワークシステムであって、ネットワークの構成変更前の論理トポロジを用いて信号伝送を行っている状態で、ネットワークの構成変更後の論理トポロジを生成し、

前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替えることを特徴とするネットワークシステム。

【請求項 8 4】 入力されるフレームが保持する宛先に関する情報と当該フレームの転送先との対応関係をスパニングツリープロトコルを用いて作成する手段と、

前記対応関係を参照し、入力されたフレームの転送先を決定する手段とを有するノード。

【発明の詳細な説明】

【 0 0 0 1 】



【発明の属する技術分野】

本発明は、ネットワークシステムに関し、特にスパニングツリーの再構成時にネットワークを停止させることが無く、さらに負荷分散機能を有するネットワークシステム、スパニングツリー構成方法及びスパニングツリー構成ノードに関する。

【0002】

【従来の技術】

従来、この種のスパニングツリーは、ループ(円環)状に形成されたネットワークにおいて、データが永遠に循環するのを防止するために用いられている。

【0003】

例えば「1998年、アイトリプルイー・スタンダード・802.1ディー (IEEE Std 802.1D)」と題するIEEE発行の標準化文書では、ループ(円環)状に形成されたネットワーク内で、データが永遠に循環するのを防止するため、ノード間でBridge Protocol Data Unit(BPDU)と呼ばれる制御情報をやり取りし、物理的にループ状になっているネットワークの一部を論理的に使用不能にして、論理的にツリー状のトポロジを形成する、スパニングツリーと呼ばれる制御手法が規定されている。これを従来技術1とする。

【0004】

また、「2001年、アイトリプルイー・スタンダード・802.1ダブリュ (IEEE Std 802.1w)」と題するIEEE発行の標準化文書では、制御情報の交換方法を拡張することにより、従来技術1におけるツリー作成を高速化し、さらに、あらかじめ迂回経路を設定しておくことにより、障害発生時の高速な迂回経路の設定を行う、高速スパニングツリーと呼ばれる制御手法が規定されている。これを従来技術2とする。

【0005】

【非特許文献1】

「1998年、アイトリプルイー・スタンダード・802.1ディー (IEEE Std 802.1D)」と題するIEEE発行の標準化文書

【非特許文献2】

「2001年、アイトリプルイー・スタンダード・802.1ダブリュ（IEEE Std 802.1w）」と題するIEEE発行の標準化文書

【0006】

【発明が解決しようとする課題】

上述した従来の技術では、以下に述べるような問題点があった。

【0007】

第1に、輻輳により、フレームの到着遅れや、欠落が発生するという問題があった。

【0008】

従来技術1では、スパニングツリーに属するノードおよびリンクの追加および削除時に、スパニングツリーを停止させて最初から構築しなおすため、再構築中に長時間ネットワーク全体が停止して輻輳が発生することにより、フレームの到着が遅れたり、欠落する場合があった。

【0009】

従来技術2では、スパニングツリーに属するノードおよびリンクの追加および削除時に、局所的にデータフレームの転送を停止させながら、徐々にスパニングツリーを構築しなおすため、再構築中にネットワークの一部が停止して輻輳し、フレームの到着が遅れたり、欠落する場合があった。

【0010】

第2に、スパニングツリーに属するノードの追加および削除等のスパニングツリー再構成時に、ネットワークが停止するという問題点があった。

【0011】

従来技術1では、スパニングツリーに属するノードの追加および削除時に、スパニングツリーを停止させて最初から構築しなおすため、再構築中に長時間ネットワーク全体が停止する場合があった。

【0012】

従来技術2では、スパニングツリーに属するノードの追加および削除時に、局所的にデータフレームの転送を停止させながら、徐々にスパニングツリーを構築しなおすため、再構築中にネットワークの一部が停止する場合があった。

【 0 0 1 3 】

第 3 に、トラフィックの負荷分散ができないという問題点があった。

【 0 0 1 4 】

従来技術 1 および従来技術 2 では、リンク容量を用いてコストを計算し、スパニングツリー構築時の経路選択に利用しているため、トラフィックに応じた動的な負荷分散のための経路変更ができなかった。

【 0 0 1 5 】

第 4 に、負荷分散しようとする、スパニングツリー再構成のために、ネットワークが停止するという問題点があった。

【 0 0 1 6 】

従来技術 1 では、コストをトラフィック状況に応じて動的に変化させようすると、スパニングツリーを一旦停止させてから構築しなおすことで経路を変更するため、再構築中に長時間ネットワーク全体が停止する場合があった。

【 0 0 1 7 】

従来技術 2 では、コストをトラフィック状況に応じて動的に変化させようすると、局所的にデータフレームの転送を停止させながら、スパニングツリーの一部を徐々に構築しなおして経路を変更するため、再構築中にネットワークの一部が停止する場合があった。

【 0 0 1 8 】

第 5 に、宛先への最低コスト経路が必ずしも選択されとは限らないという問題があった。

【 0 0 1 9 】

従来技術 1 および従来技術 2 では、スパニングツリーをネットワーク上で 1 系統のみ設定し、各ノード毎に予め設定されたプライオリティ値および MAC アドレスより、ネットワーク上で唯一のルートノードを決定し、単一のツリーを作成するため、ツリーの末端に位置するノード同士が通信する場合は、最短経路が別にあったとしてもブロッキングされてしまい、冗長な経路を通過する場合があった。

【 0 0 2 0 】

第 6 に、リンク利用率が低い一方、ルートノード付近に負荷が集中するという問題があった。

## 【 0 0 2 1 】

従来技術 1 および従来技術 2 では、スパニングツリーをネットワーク上で 1 系統のみ設定し、各ノードに設定されたプライオリティ値および MAC アドレスより、ネットワーク上で唯一のルートノードを決定し、単一のツリーを作成するため、ツリーの末端では設置されていても利用されないリンクが出現してリンク利用率が低下した。逆に、ルートノード付近では、トラフィックが集中し、輻輳発生の可能性が高まる場合があった。

## 【 0 0 2 2 】

第 7 に、ルートノード障害時のツリー構築に時間がかかり、その間ネットワークが停止するという問題があった。

## 【 0 0 2 3 】

従来技術 1 では、スパニングツリーをネットワーク上で 1 系統のみ設定し、ルートノードを 1 つだけしか持たないため、ルートノードにおいて障害が発生すると、スパニングツリーを停止させて最初から構築しなおすため、再構築中に長時間ネットワーク全体が停止する場合があった。

## 【 0 0 2 4 】

従来技術 2 では、ルートノードにおいて障害が発生すると、局所的にデータフレームの転送を停止させながら、徐々にスパニングツリーを構築しなおすため、再構築中にネットワークの一部が停止する場合があった。

## 【 0 0 2 5 】

第 8 に、IEEE 802.1D を利用する区間は、障害時のルート切り替えが遅く、スパニングツリーの再構成にも時間がかかるという問題があった。

## 【 0 0 2 6 】

従来技術 1 では、ツリーの構築時においてデータの交換が可能となるまで、数十秒の時間がかかる場合があるからである。

## 【 0 0 2 7 】

さらに、第 9 に、従来技術 1 及び従来技術 2 の何れにおいても、単一のツリー

しか持たないために、トラフィックがルートノード付近に集中して輻輳し、フレームの到着が遅れたり、欠落する場合があった。

【 0 0 2 8 】

本発明の第 1 の目的は、輻輳発生確率を下げ、輻輳によるフレームの到着遅れや欠落が発生する頻度を減らすことができる、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供することにある。

【 0 0 2 9 】

本発明の第 2 の目的は、ネットワークを停止させずに、スパニングツリーに属するノードの追加および削除等のスパニングツリー再構成ができる、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供することにある。

【 0 0 3 0 】

本発明の第 3 の目的は、トラフィックの負荷分散ができる、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供することにある。

【 0 0 3 1 】

本発明の第 4 の目的は、経路変更にとまなうスパニングツリー再構成のために、ネットワークを停止させることなく負荷分散ができる、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供することにある。

【 0 0 3 2 】

本発明の第 5 の目的は、宛先への最低コスト経路が選択される、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供することにある。

【 0 0 3 3 】

本発明の第 6 の目的は、リンク利用率を高め、ルートノード付近に負荷を集中させずに分散可能な、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供すること

にある。

【 0 0 3 4 】

本発明の第 7 の目的は、ルートノード障害によるネットワーク停止を回避可能な、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供することにある。

【 0 0 3 5 】

本発明の第 8 の目的は、スパニングツリーが I E E E 8 0 2 . 1 D 利用区間を通過して設定されるのを防ぎ、障害時の切替やルート変更を高速化し、輻輳発生の可能性やフレームの欠落の可能性を低下させることが可能な、ネットワークシステム、スパニングツリー構成方法、スパニングツリー構成ノード、及びスパニングツリー構成プログラムを提供することにある。

【 0 0 3 6 】

【課題を解決するための手段】

請求項 1 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とする。

【 0 0 3 7 】

請求項 2 の本発明のノードは、前記ネットワークの構成変更が、ノードの追加又は削除やリンクトポロジの変化であることを特徴とする。

【 0 0 3 8 】

請求項 3 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とする。

【 0 0 3 9 】

請求項 4 の本発明は、複数のノードを接続したネットワーク上でスパニングツ



リーを構成するノードにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、転送に利用しているスパニングツリーに対応したタグを返答するタグテーブルと、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、転送に利用するスパニングツリーを決定するツリーセクタと、宛先ごとのフレームの転送出力先を記録するフォワーディングテーブルと、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器とを備えることを特徴とする。

## 【 0 0 4 0 】

請求項 5 の本発明のノードは、前記ツリーセクタが、転送に利用するスパニングツリーの切り替えを行うメインコントローラと、スパニングツリーの安定を示す指定した時間のタイマ満了を知らせる安定タイマと、フレームに付加されたタグを削除するタグ削除器と、スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信器と、フレームにタグを付加するタグ挿入器とを備えることを特徴とする。

## 【 0 0 4 1 】

請求項 6 の本発明のノードは、前記ツリーセクタは、スパニングツリーの安定を示すフレームの到着間隔を判別するため、一定時間が経過するとタイマ満了通知を送信する到着間隔タイマを備えることを特徴とする。

## 【 0 0 4 2 】

請求項 7 の本発明のノードは、前記ツリーセクタは、リンクコストの算出に利用する指定時間のタイマ満了を知らせるコスト参照タイマを備えることを特徴とする。

## 【 0 0 4 3 】

請求項 8 の本発明のノードは、前記ツリーマネージャが、フレームに付加されたタグを削除するタグ削除器と、 B P D U を送受信する B P D U 送受信器と、フレームにタグを付加するタグ挿入器と、スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコントローラと、前記スパニングツリープロトコルで使用するパラメータを保持するツリーテーブルとを備えることを特徴とす

る。

【 0 0 4 4 】

請求項 9 の本発明のノードは、前記ツリーマネージャは、通知されたリンクコストに所定の設定値を加算して返答するコスト操作器を備えることを特徴とする。

【 0 0 4 5 】

請求項 1 0 の本発明のノードは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタを備えることを特徴とする。

【 0 0 4 6 】

請求項 1 1 の本発明のノードは、前記リンクコストを、利用状況に基づいて計算することを特徴とする。

【 0 0 4 7 】

請求項 1 2 の本発明のノードは、前記利用状況を、空き帯域と定義することを特徴とする。

【 0 0 4 8 】

請求項 1 3 の本発明のノードは、前記利用状況を、CPU 負荷と定義することを特徴とする。

【 0 0 4 9 】

請求項 1 4 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、ネットワーク内の各ノードがルートノードとなるスパニングツリーを生成し、宛先がルートノードとなるスパニングツリーを用いてフレームを転送することを特徴とする。

【 0 0 5 0 】

請求項 1 5 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、転送に利用しているスパニングツリーに対応したタグを返答するタグテーブルと、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、ネットワークに存在するルートノード数と同数のツリーマネージャを生成するツリーセクタと、宛先ごとのフレームの転



送出力先を記録するフォワーディングテーブルと、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器とを備えることを特徴とする。

## 【 0 0 5 1 】

請求項 1 6 の本発明のノードは、前記ツリーセクタが、ツリーマネージャの作成もしくは削除を行うメインコントローラと、フレームに付加されたタグを削除するタグ削除器と、スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信器と、フレームにタグを付加するタグ挿入器とを備えることを特徴とする。

## 【 0 0 5 2 】

請求項 1 7 の本発明のノードは、前記ツリーマネージャが、フレームに付加されたタグを削除するタグ削除器と、B P D U を送受信する B P D U 送受信器と、フレームにタグを付加するタグ挿入器と、スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコントローラと、前記スパニングツリープロトコルで使用するパラメータを保持するツリーテーブルとを備えることを特徴とする。

## 【 0 0 5 3 】

請求項 1 8 の本発明のノードは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタを備えることを特徴とする。

## 【 0 0 5 4 】

請求項 1 9 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、スパニングツリーを生成するツリーマネージャが、スパニングツリープロトコルの種類およびバージョンにより、コスト値を調整するコスト操作器を備えることを特徴とする。

## 【 0 0 5 5 】

請求項 2 0 の本発明のノードは、前記コスト操作器は、障害回復処理が遅いプロトコルを利用しているリンクに対して、大きなコストを割り振ることを特徴とする。

## 【 0 0 5 6 】

請求項 2 1 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、ネットワーク内の動作の遅いプロトコルを使用中のリンクのそれぞれについて、前記各リンクのコストが最大となるスパニングツリーを生成し、前記各リンクに障害が発生した際に、前記リンクのコストが最大となっているツリーを利用してフレーム転送を行うことを特徴とする。

## 【 0 0 5 7 】

請求項 2 2 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成するノードにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、転送用ツリーに対応したタグを返答するタグテーブルと、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、ネットワークに存在する動作の遅いプロトコルを使用中のリンク数と同数のツリーマネージャを生成するツリーセクタと、宛先ごとのフレーム転送出力先を記録するフォワーディングテーブルと、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、前記タグに従って転送先のツリーマネージャを決定する分別器とを備えることを特徴とする。

## 【 0 0 5 8 】

請求項 2 3 の本発明のノードは、前記ツリーセクタが、ツリーマネージャの作成もしくは削除を行うツリーセクタ内のメインコントローラと、フレームに付加されたタグを削除するタグ削除器と、制御フレームを送信する G V R P 送受信器と、フレームにタグを付加するタグ挿入器とを備えることを特徴とする。

## 【 0 0 5 9 】

請求項 2 4 の本発明のノードは、前記ツリーマネージャが、フレームに付加されたタグを削除するタグ削除器と、 B P D U を送受信する B P D U 送受信器と、フレームにタグを付加するタグ挿入器と、スパニングツリープロトコルにしたがってスパニングツリーを作成するツリーコントローラと、スパニングツリープロトコルで使用するパラメータを保持するツリーテーブルとを備えることを特徴とする。

【 0 0 6 0 】

請求項 2 5 の本発明のノードは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタを備えることを特徴とする。

【 0 0 6 1 】

請求項 2 6 の本発明のノードは、スパニングツリープロトコルで利用されている H E L L O フレームより、短い間隔で障害検出用フレームを送受信して障害を検出する障害検出器を備えることを特徴とする。

【 0 0 6 2 】

請求項 2 7 の本発明のノードは、前記フォワーディングテーブルが、ブロードキャスト出力ポートフィールドを有することを特徴とする。

【 0 0 6 3 】

請求項 2 8 の本発明のノードは、前記フォワーディングテーブルが、予備出力ポートフィールドを有することを特徴とする。

【 0 0 6 4 】

請求項 2 9 の本発明のノードは、スパニングツリーによって決定されるポート種別を用いて、出力先ポートを決定することを特徴とする。

【 0 0 6 5 】

請求項 3 0 の本発明のノードは、前記スパニングツリーによって決定されるポート種別が、ルートポート及びディジグネイテッドポートの何れかであることを特徴とする。

【 0 0 6 6 】

請求項 3 1 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替える処理を実行することを特徴とする。

【 0 0 6 7 】

請求項 3 2 の本発明のスパニングツリー構成プログラムは、前記ネットワーク

の構成変更が、ノードの追加又は削除やリンクトポロジの変化であることを特徴とする。

## 【 0 0 6 8 】

請求項 3 3 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替える処理を実行することを特徴とする。

## 【 0 0 6 9 】

請求項 3 4 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、複数のツリーマネージャによる独立して動作する複数のスパニングツリーを生成する処理と、転送に利用しているスパニングツリーに対応したタグを返答する処理と、返答のあった前記タグをフレームに挿入するタグ挿入処理と、転送に利用するツリーを決定するツリーセクタ処理と、宛先ごとのフレームの転送出力先を記録するフォワーディングテーブル処理と、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送処理と、前記タグに従って転送先のツリーマネージャを決定する分別処理とを実行することを特徴とする。

## 【 0 0 7 0 】

請求項 3 5 の本発明のスパニングツリー構成プログラムは、前記ツリーセクタ処理において、転送に利用するスパニングツリーの切り替えを行うコントローラ処理と、スパニングツリーの安定を示す指定した時間のタイマ満了を知らせる安定タイマ処理と、フレームに付加されたタグを削除するタグ削除処理と、スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信処理と、フレームにタグを付加するタグ挿入処理とを実行することを特徴とする。

## 【 0 0 7 1 】

請求項 3 6 の本発明のスパニングツリー構成プログラムは、前記ツリーセクタ

タ処理において、スパニングツリーの安定を示すフレームの到着間隔を判別するため、一定時間が経過すると、タイマ満了通知を送信する到着間隔タイマ処理を実行することを特徴とする。

## 【 0 0 7 2 】

請求項 3 7 の本発明のスパニングツリー構成プログラムは、前記ツリーセレクト処理は、リンクコストの算出に利用する指定時間のタイマ満了を知らせるコスト参照タイマ処理を実行することを特徴とする。

## 【 0 0 7 3 】

請求項 3 8 の本発明のスパニングツリー構成プログラムは、前記ツリーマネージャ処理が、フレームに付加されたタグを削除するタグ削除処理と、B P D U を送受信する B P D U 送受信処理と、フレームにタグを付加するタグ挿入処理と、スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコントローラ処理と、前記スパニングツリープロトコルで使用するパラメータを保持するツリーテーブル処理とを実行することを特徴とする。

## 【 0 0 7 4 】

請求項 3 9 の本発明のスパニングツリー構成プログラムは、前記ツリーマネージャ処理は、通知されたリンクコストに所定の設定値を加算して返答するコスト操作処理を実行することを特徴とする。

## 【 0 0 7 5 】

請求項 4 0 の本発明のスパニングツリー構成プログラムは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタ処理を実行することを特徴とする。

## 【 0 0 7 6 】

請求項 4 1 の本発明のスパニングツリー構成プログラムは、リンクコストを、利用状況により計算する処理を実行することを特徴とする。

## 【 0 0 7 7 】

請求項 4 2 の本発明のスパニングツリー構成プログラムは、前記利用状況を、空き帯域と定義することを特徴とする。

## 【 0 0 7 8 】

請求項 4 3 の本発明のスパニングツリー構成プログラムは、前記利用状況を、CPU 負荷と定義することを特徴とする。

## 【 0 0 7 9 】

請求項 4 4 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、ネットワーク内の各ノードがルートノードとなるスパニングツリーを生成し、宛先がルートノードとなるツリーを用いてフレームを転送することを特徴とする。

## 【 0 0 8 0 】

請求項 4 5 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノード上で動作するスパニングツリー構成プログラムにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャ処理と、転送に利用しているスパニングツリーに対応したタグを返答するタグテーブル処理と、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入処理と、ネットワークに存在するルートノード数と同数のツリーマネージャを生成するツリーセクタ処理と、宛先ごとのフレーム転送出力先を記録するフォワーディングテーブル処理と、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送処理と、前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別処理とを備えることを特徴とする。

## 【 0 0 8 1 】

請求項 4 6 の本発明のスパニングツリー構成プログラムは、前記ツリーセクタ処理が、ツリーマネージャの作成もしくは削除を行うツリーセクタ内のメインコントローラ処理と、フレームに付加されたタグを削除するタグ削除処理と、スパニングツリーの切り替えのための制御フレームを送信する G V R P 送受信処理と、フレームにタグを付加するタグ挿入処理とを実行することを特徴とする。

## 【 0 0 8 2 】

請求項 4 7 の本発明のスパニングツリー構成プログラムは、前記ツリーマネージャ処理が、フレームに付加されたタグを削除するタグ削除処理と、B P D U を



送受信する B P D U 送受信処理と、フレームにタグを付加するタグ挿入処理と、スパニングツリープロトコルに従ってスパニングツリーを作成するツリーコントローラ処理と、前記スパニングツリープロトコルで使用するパラメータを保持するツリーテーブル処理とを実行することを特徴とする。

## 【 0 0 8 3 】

請求項 4 8 の本発明のスパニングツリー構成プログラムは、各前記ノードは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタ処理を実行することを特徴とする。

## 【 0 0 8 4 】

請求項 4 9 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノードで動作するスパニングツリー構成プログラムにおいて、ネットワーク内の動作の遅いプロトコルを使用中のリンクのそれぞれについて、前記各リンクのコストが最大となるスパニングツリーを生成し、前記各リンクに障害が発生した際に、前記リンクのコストが最大となっているツリーを利用してフレーム転送を行うことを特徴とする。

## 【 0 0 8 5 】

請求項 5 0 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーを構成する各ノードで動作するスパニングツリー構成プログラムにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャ処理と、転送用ツリーに対応したタグを返答するタグテーブル処理と、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入処理と、ネットワークに存在する動作の遅いプロトコルを使用中のリンク数と同数のツリーマネージャを生成するツリーセクタ処理と、宛先ごとのフレームの転送出力先を記録するフォワーディングテーブル処理と、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送処理と、前記タグに従ってフレームの転送先のツリーマネージャを決定する分別処理とを備えることを特徴とする。

## 【 0 0 8 6 】

請求項 5 1 の本発明のスパニングツリー構成プログラムは、前記ツリーセクタ

タ処理が、ツリーマネージャの作成もしくは削除を行うツリーセクタ内のメインコントローラ処理と、フレームに付加されたタグを削除するタグ削除処理と、制御フレームを送信するGVRP送受信処理と、フレームにタグを付加するタグ挿入処理とを備えることを特徴とする。

【 0 0 8 7 】

請求項 5 2 の本発明のスパニングツリー構成プログラムは、前記ツリーマネージャ処理が、フレームに付加されたタグを削除するタグ削除処理と、BPDUを送受信するBPDU送受信処理と、フレームにタグを付加するタグ挿入処理と、スパニングツリープロトコルにしたがってスパニングツリーを作成するツリーコントローラ処理と、スパニングツリープロトコルで使用するパラメータを保持するツリーテーブル処理とを備えることを特徴とする。

【 0 0 8 8 】

請求項 5 3 の本発明のスパニングツリー構成プログラムは、各前記ノードは、リンクの接続状況や空き帯域を含むリソース情報を計測するリソースモニタ処理を実行することを特徴とする。

【 0 0 8 9 】

請求項 5 4 の本発明のスパニングツリー構成プログラムは、前記フォワーディングテーブルが、ブロードキャスト出力ポートフィールドを有することを特徴とする。

【 0 0 9 0 】

請求項 5 5 の本発明のスパニングツリー構成プログラムは、前記フォワーディングテーブルが、予備出力ポートフィールドを有することを特徴とする。

【 0 0 9 1 】

請求項 5 6 の本発明のスパニングツリー構成プログラムは、スパニングツリーによって決定されるポート種別を用いて、出力先ポートを決定することを特徴とする。

【 0 0 9 2 】

請求項 5 7 の本発明のスパニングツリー構成プログラムは、前記スパニングツリーによって決定されるポート種別が、ルートポート及びディジグネイテッドポ



ートの何れかであることを特徴とする。

【 0 0 9 3 】

請求項 5 8 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、各前記ノードが、ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とする。

【 0 0 9 4 】

請求項 5 9 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、各前記ノードが、ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とする。

【 0 0 9 5 】

請求項 6 0 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、各前記ノードが、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、転送に利用しているスパニングツリーに対応したタグを返答するタグテーブルと、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、転送に利用するスパニングツリーを決定するツリーセレクタと、宛先ごとのフレームの転送出力先を記録するフォワーディングテーブルと、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器とを備えることを特徴とする。

【 0 0 9 6 】

請求項 6 1 の本発明のネットワークシステムは、リンクコストを、利用状況により計算することを特徴とする。

## 【 0 0 9 7 】

請求項 6 2 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、ネットワーク内の各ノードがルートノードとなるスパニングツリーを生成し、宛先がルートノードとなるツリーを用いてフレームを転送することを特徴とする。

## 【 0 0 9 8 】

請求項 6 3 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、転送用ツリーに対応したタグを返答するタグテーブルと、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、ネットワークに存在するノード数と同数のツリーマネージャを生成するツリーセレクトと、宛先ごとのフレーム転送出力先を記録するフォワーディングテーブルと、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器とを備えることを特徴とする。

## 【 0 0 9 9 】

請求項 6 4 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、スパニングツリーを生成するツリーマネージャが、スパニングツリープロトコルの種類およびバージョンにより、コスト値を調整するコスト操作処理を実行することを特徴とする。

## 【 0 1 0 0 】

請求項 6 5 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、スパニングツリーを生成するツリーマネージャが、スパニングツリープロトコルの種類およびバージョンにより、コスト値を調整するコスト操作器を備えることを特徴とする。

## 【 0 1 0 1 】

請求項 6 6 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、ネットワーク内の動作の遅いプロトコルを使用中のリンクのそれぞれについて、前記各リンクのコストが最大となるスパニングツリーを生成し、前記各リンクに障害が発生した際に、前記リンクのコストが最大となっているツリーを利用してフレーム転送を行うことを特徴とする。

## 【 0 1 0 2 】

請求項 6 7 の本発明は、複数のノードを接続したネットワーク上でスパニングツリーによって転送経路を設定するネットワークシステムにおいて、独立して動作する複数のスパニングツリーを生成する複数のツリーマネージャと、転送用ツリーに対応したタグを返答するタグテーブルと、前記タグテーブルより返答のあったタグをフレームに挿入するタグ挿入器と、ネットワークに存在する動作の遅いプロトコルを使用中のリンク数と同数のツリーマネージャを生成するツリーセレクタと、宛先ごとのフレーム転送出力先を記録するフォワーディングテーブルと、前記フォワーディングテーブルで指定された転送出力先にフレームを転送するフレーム転送器と、前記タグに従って前記フレームの転送先のツリーマネージャを決定する分別器とを備えることを特徴とする。

## 【 0 1 0 3 】

請求項 6 8 の本発明のネットワークシステムは、前記フォワーディングテーブルが、ブロードキャスト出力ポートフィールドを有することを特徴とする。

## 【 0 1 0 4 】

請求項 6 9 の本発明のネットワークシステムは、前記フォワーディングテーブルが、予備出力ポートフィールドを有することを特徴とする。

## 【 0 1 0 5 】

請求項 7 0 の本発明のネットワークシステムは、スパニングツリーによって決定されるポート種別を用いて、出力先ポートを決定することを特徴とする。

## 【 0 1 0 6 】

請求項 7 1 の本発明のネットワークシステムは、前記スパニングツリーによって決定されるポート種別が、ルートポート及びディジグネイテッドポートの何れ

かであることを特徴とする。

【 0 1 0 7 】

請求項 7 2 の本発明は、複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、ネットワークの構成変更前のスパニングツリーを運用したまま、構成変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とする。

【 0 1 0 8 】

請求項 7 3 の本発明は、複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、ネットワークのコスト変更時に、現スパニングツリーを運用したまま、リンクコスト変更後の新たなスパニングツリーを生成し、前記新たなスパニングツリーの安定後に、転送に利用するスパニングツリーを前記新たなスパニングツリーに切り替えることを特徴とする。

【 0 1 0 9 】

請求項 7 4 の本発明は、複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、新規ノードを追加する際に、新規ノードを既存スパニングツリーに参加させず、予備スパニングツリーにのみ参加させることを特徴とする。

【 0 1 1 0 】

請求項 7 5 の本発明は、複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、ノードを削除する際に、削除ノードを予備スパニングツリーに参加させず、既存スパニングツリーのみに参加させることを特徴とする。

【 0 1 1 1 】

請求項 7 6 の本発明は、複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、ネットワーク構成が変更された時、予備系を用いて変更後のツリー作成を行うことを特徴とする。

【 0 1 1 2 】

請求項 7 7 の本発明は、複数のノードを接続したネットワークにおけるスパニ

ングツリー構成方法であって、コスト算出に、リンクの空き帯域を利用することを特徴とする。

【 0 1 1 3 】

請求項 7 8 の本発明は、複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、ネットワーク内の全ノードが、全ノードをメンバに持つスパニングツリーのうち、どれか 1 つのルートノードとなるよう、複数のスパニングツリーを作成することを特徴とする。

【 0 1 1 4 】

請求項 7 9 の本発明は、複数のノードを接続したネットワークにおけるスパニングツリー構成方法であって、ネットワーク内に存在する全ノードをメンバに持つスパニングツリーを作成し、うち、障害復旧の遅いプロトコルを使用しているリンクごとに複数のスパニングツリーを作成することを特徴とする。

【 0 1 1 5 】

請求項 8 0 の本発明は、複数のノードを接続したネットワークにおいて信号伝送に用いる論理トポロジを形成する方法であって、ネットワークの構成変更前の論理トポロジを用いて信号伝送を行っている状態で、ネットワークの構成変更後の論理トポロジを生成し、前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替えることを特徴とする。

【 0 1 1 6 】

請求項 8 1 の本発明のノードは、自らが属するネットワークにおいて信号伝送に用いる論理トポロジを用いて信号伝送を行っている状態で、前記ネットワークの構成変更時に、ネットワークの構成変更後の論理トポロジを生成する手段と、前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替える手段とを有する。

【 0 1 1 7 】

請求項 8 2 の本発明のプログラムは、自らが属するネットワークにおいて信号伝送に用いる論理トポロジを用いて信号伝送を行っている状態で、前記ネットワークの構成変更時に、ネットワークの構成変更後の論理トポロジを生成する処理

と、前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替える処理とを実行することを特徴とする。

【 0 1 1 8 】

請求項 8 3 の本発明は、複数のノードを接続したネットワークシステムであって、ネットワークの構成変更前の論理トポロジを用いて信号伝送を行っている状態で、ネットワークの構成変更後の論理トポロジを生成し、前記構成変更後の論理トポロジ安定後に、信号伝送に用いる論理トポロジを前記構成変更後の論理トポロジに切り替えることを特徴とする。

【 0 1 1 9 】

請求項 8 4 の本発明のノードは、入力されるフレームが保持する宛先に関する情報と当該フレームの転送先との対応関係をスパニングツリープロトコルを用いて作成する手段と、前記対応関係を参照し、入力されたフレームの転送先を決定する手段とを有する。

【 0 1 2 0 】

【発明の実施の形態】

以降の説明の中で、複数のスパニングツリーおよび複数のノード群を識別する識別子としてタグを用いた説明を行うが、このタグは、VLANタグのほかに、本特許出願人による特願 2 0 0 2 - 2 0 4 6 7 3 号に開示される拡張タグ及び他のタグもしくは識別手段のうち、単独もしくは何れか 1 つ以上の組み合わせを意味する。

【 0 1 2 1 】

ここで、本発明において用いるタグのうち、上記特願 2 0 0 2 - 2 0 4 6 7 3 号に開示される拡張タグ付きフレームのフォーマットについて説明する。

【 0 1 2 2 】

図 1 は、IEEE 8 0 2 . 1 Q で規定されている VLAN タグ付きのイーサネット(R)フレームのフォーマットである。VLAN タグ付きイーサネット(R)フレーム 3 2 0 0 は、送信先 MAC アドレス 3 2 0 1 と、送信元 MAC アドレス 3 2 0 2 と、VLAN タグ 3 2 0 3 と、イーサネット(R)属性情報 3 2 0 4 と、ペイ



ロード 3 2 0 5 と、F C S 3 2 0 6 とから構成される。

【 0 1 2 3 】

これに対して、図 2 は、本発明の拡張タグ付きイーサネット(R)フレームのフォーマットである。拡張タグ付きイーサネット(R)フレーム 3 3 0 0 は、送信先 MAC アドレス 3 2 0 1 と、送信元 MAC アドレス 3 2 0 2 と、拡張タグ格納領域 3 3 0 1 と、イーサネット(R)属性情報 3 2 0 4 と、ペイロード 3 2 0 5 と、F C S 3 2 0 6 とから構成され、既存の V L A N タグ付きイーサネット(R)フレーム 3 2 0 0 の V L A N タグ 3 2 0 3 が拡張タグ格納領域 3 3 0 1 に置き換わる。

【 0 1 2 4 】

また図 3 に示すように別の構成の拡張タグ付きイーサネット(R)フレーム 3 4 0 0 も存在し、これは送信先 MAC アドレス 3 2 0 1 と、送信元 MAC アドレス 3 2 0 2 と、拡張タグ格納領域 3 3 0 1 と、V L A N タグ 3 2 0 3 と、イーサネット(R)属性情報 3 2 0 4 と、ペイロード 3 2 0 5 と、F C S 3 2 0 6 とから構成され、拡張タグ格納領域 3 3 0 1 は送信元 MAC アドレス 3 2 0 2 の後に挿入される。

【 0 1 2 5 】

拡張タグ格納領域 3 3 0 1 には 1 つまたは複数の拡張タグを格納可能である。拡張タグのサイズは 4 バイトであり、V L A N タグ 3 2 0 3 と同一サイズとなっている。拡張タグ付きイーサネット(R)フレーム 3 3 0 0、3 4 0 0 の最上段の拡張タグと V L A N タグ付きイーサネット(R)フレーム 3 2 0 0 の V L A N タグは同一の位置に同一のサイズで格納されており、その区別は各々のタグの上位 2 バイトに格納される値を変更することにより区別する（詳細については後述する）。

【 0 1 2 6 】

これにより、拡張タグ付きイーサネット(R)フレーム 3 3 0 0、3 4 0 0 は、V L A N タグ付きイーサネット(R)フレーム 3 2 0 0 と互換性を有しており、既存ノード、拡張タグ対応ノードの双方においてどちらのフレームも処理可能である。

## 【 0 1 2 7 】

図 4 は、拡張タグ格納領域 3 3 0 1 を示している。図 4 に示す格納例では、8 個の拡張タグ 3 5 0 0 ～ 3 5 0 7 が格納されている。

## 【 0 1 2 8 】

フォワーディングタグ 3 5 0 0 には宛先ノードの識別子や宛先までのラベル（例えば M P L S ラベル）が格納される。また、宛先ノードの識別子を格納したフォワーディングタグ 3 5 0 0 に加えて、送信元ノードの識別子を格納する場合もある。各ノードはこのフォワーディングタグを参照してフレーム転送先を決定する。このフォワーディングタグ 3 5 0 0 は拡張タグ付きイーサネット(R)フレーム 3 3 0 0、3 4 0 0 において必ず格納される。

## 【 0 1 2 9 】

拡張タグの種類としては、カスタマ分離タグ 3 5 0 1、プロテクションタグ 3 5 0 2、O A M & P タグ 3 5 0 3、品質情報タグ 3 5 0 4、フレーム制御タグ 3 5 0 5、セキュリティタグ 3 5 0 6、ユーザ拡張タグ 3 5 0 7 が格納されている。

## 【 0 1 3 0 】

カスタマ分離タグ 3 5 0 2 には、各ノードに収容されるカスタマ毎に情報を分離するための識別子が格納される。カスタマとしては、同一の V L A N の所属するカスタマを同一カスタマとする場合や、2 つ以上のノードの特定のポートに収容されるカスタマを同一カスタマとする場合や、網内のノードに接続する 2 つ以上のホストを同一カスタマとする場合などがある。これらのカスタマに対して分離識別子が割り当てられ、各カスタマからのフレームにはカスタマ分離タグ 3 5 0 2 内にこの分離識別子が格納される。カスタマ分離タグ 3 5 0 2 によってカスタマを識別することにより、カスタマ単位の付加サービス（例えば、特定カスタマに対する優先制御など）の提供が可能となる。また、カスタマ分離タグ 3 5 0 2 は複数スタックして使うことも可能である。この場合、分離可能なカスタマ数を大幅に増やすことができる。なお、カスタマ分離タグ 3 5 0 2 をスタックする場合には、スタックした最終段のカスタマ分離タグ 3 5 0 2 は最終段であることを示す特別なカスタマ分離タグを使用する。

## 【 0 1 3 1 】

プロテクションタグ 3 5 0 3 には、障害発生時の障害情報や障害復旧のための迂回経路情報が格納される。OAM&P タグ 3 5 0 4 には、運用／管理情報が格納される。

## 【 0 1 3 2 】

品質情報タグ 3 5 0 5 には、遅延、ジッタ、パケットロス率やフレームのネットワーク内への流入時間を示すタイムスタンプ、帯域制御情報等の品質情報が格納される。品質情報タグ 3 5 0 5 にタイムスタンプ値が格納されている場合、フレームを受信したノードは現在の時刻とタイムスタンプ値よりそのフレームの網内遅延（ネットワーク内での滞在時間）を算出することができる。網内遅延の保証値が定められている場合には、保証値を実現できるよう優先処理を実施することができる。また、品質情報タグ 3 5 0 5 に要求帯域や蓄積データ量やトラヒッククラス等の帯域制御情報が格納されている場合、そのフローの蓄積データ量やトラヒッククラスと他のフローのトラヒック状況を考慮して、要求帯域を確保するための帯域制御を実施することができる。

## 【 0 1 3 3 】

フレーム制御タグ 3 5 0 6 には、フレームのネットワーク内での生存時間を制限するホップカウンタ（T T L : Time To Live）や誤り検出のための C R C などの情報が格納される。T T L が格納される場合には、経由するノード毎に T T L 値が減算され、T T L = 0 となったところでこのフレームは廃棄される。これにより、ループ経路になった場合でもフレームが循環し続けるのを防ぐことができる。また C R C が格納される場合には、入口側ノードにおける拡張タグ格納領域 3 3 0 1 の C R C 演算結果が格納されており、出口側ノードにおいて再度 C R C 演算を実施して格納値と比較することにより、拡張タグ格納領域 3 3 0 1 の誤りを検出可能である。

## 【 0 1 3 4 】

セキュリティタグ 3 5 0 7 には、フレームの信頼性、ネットワーク構築時やネットワーク構成変更時の秘匿性を確保するための情報が格納される。セキュリティタグ 3 5 0 7 の利用例としては以下の例があげられる。ネットワーク内で通信

するカスタマにはカスタマ毎のセキュリティ識別子が予め設定され、その識別子はカスタマが接続する各ノードで保持される。各カスタマはフレーム転送の際に、設定されたセキュリティ識別子を常にセキュリティタグ 3 5 0 7 に格納することにより、カスタマ分離タグ 3 5 0 1 の情報を改ざんした悪意のあるカスタマからのフレーム送受信を防ぐことができる。また、ネットワーク構築時やネットワーク構成変更時において、ノード間でネゴシエートして共通のセキュリティ識別子を設定する。当該ノード間のフレーム転送の際に、設定したセキュリティ識別子を常にセキュリティタグ 3 5 0 7 に格納することにより、悪意のあるノードをネットワークに接続するのを防ぐことができる。

#### 【 0 1 3 5 】

ユーザ拡張タグ 3 5 0 7 には、ユーザが独自に定義する任意の情報が格納される。ユーザが独自にタグのフォーマット及び格納情報を定義し、その処理内容を定義することにより、ユーザ独自の機能拡張を図ることができ、ネットワークの柔軟性を高められる。

#### 【 0 1 3 6 】

フォワーディングタグ 3 5 0 0 以外の拡張タグ 3 5 0 1 ~ 3 5 0 7 は、必要に応じて格納される。フォワーディングタグ 3 5 0 0 は、拡張タグ格納領域 3 3 0 1 の先頭に格納され、その他の拡張タグ 3 5 0 1 ~ 3 5 0 7 はその後方に格納される。フォワーディングタグ 3 5 0 0 よりも後方であれば、予め決められた固定位置でも、任意の位置でも良い。

#### 【 0 1 3 7 】

以降、2 系統存在するスパニングツリーのうち、新しくネットワークに侵入するデータフレームの転送に利用しているスパニングツリーを現用ツリーもしくは現用系ツリー、現用ツリーとなっていないツリーを予備用ツリーもしくは予備系ツリーと表現する。

#### 【 0 1 3 8 】

また、現用系ツリーの生成を行うツリーマネージャを現用系ツリーマネージャ、予備系ツリーの生成を行うツリーマネージャを、予備系ツリーマネージャと呼ぶ。

【 0 1 3 9 】

タググループとは、タグおよびその他の識別子を利用して識別されるノード群、すなわち複数のノードの集合体をいう。タググループをVLANタグを識別子として形成した場合は、前記タググループをVLANと呼ぶ。

【 0 1 4 0 】

B P D U (Bridge Protocol Data Unit)とは、スパニングツリー生成のために交換されるI E E E 8 0 2 . 1 D (従来技術1) およびI E E E 8 0 2 . 1 w (従来技術2) に記載される制御データおよび、本発明の現用系と予備系の識別情報等を含む制御フレームのことをいう。

【 0 1 4 1 】

図5は、I E E E 8 0 2 . 1 D (従来技術1) およびI E E E 8 0 2 . 1 w (従来技術2) に記載の、C o n f i g u r a t i o n B P D Uフレーム2205の構造を示すフォーマット図である。

【 0 1 4 2 】

M A C D A 2 2 0 1 は、宛先M A C アドレスを格納する領域である。

【 0 1 4 3 】

M A C S A 2 2 0 2 は、送信元M A C アドレスを格納する領域である。

【 0 1 4 4 】

タグ領域2203は、複数のスパニングツリーを識別する識別子としてのタグを挿入する領域である。また、従来技術には記載されていないが、前記タグは、VLANタグのほかに、本特許出願人による特願2002-204673号に開示される拡張タグ、及び他のタグもしくは識別手段の何れか1つ以上の組み合わせたタグでもよい。

【 0 1 4 5 】

T y p e 2 2 0 4 は、フレームのタイプ識別子を格納する領域である。

【 0 1 4 6 】

B P D U 領域2205は、I E E E 8 0 2 . 1 D (従来技術1) およびI E E E 8 0 2 . 1 w (従来技術2) に記載の、C o n f i g u r a t i o n B P D U p a r a m e t e r s に相当する情報を格納する領域である。

【 0 1 4 7 】

F C S 2 2 0 6 は、フレームチェックシーケンスを格納する領域である。

【 0 1 4 8 】

P r o t o c o l I d e n t i f i e r 2 2 0 5 1 は、I E E E 8 0 2 . 1 D (従来技術 1) もしくは I E E E 8 0 2 . 1 w (従来技術 2) に記載の、P r o t o c o l I d e n t i f i e r と等しい情報を格納する領域である。

【 0 1 4 9 】

P r o t o c o l V e r s i o n I d e n t i f i e r 2 2 0 5 2 は、I E E E 8 0 2 . 1 D (従来技術 1) もしくは I E E E 8 0 2 . 1 w (従来技術 2) に記載の、P r o t o c o l V e r s i o n I d e n t i f i e r と等しい情報を格納する領域である。

【 0 1 5 0 】

B P D U T y p e 2 2 0 5 3 は、I E E E 8 0 2 . 1 D (従来技術 1) もしくは I E E E 8 0 2 . 1 w (従来技術 2) に記載の、B P D U T y p e と等しい情報を格納する領域である。

【 0 1 5 1 】

F l a g s 2 2 0 5 4 は、I E E E 8 0 2 . 1 D (従来技術 1) もしくは I E E E 8 0 2 . 1 w (従来技術 2) に記載の、F l a g s と等しい情報を格納する領域である。

【 0 1 5 2 】

R o o t I d e n t i f i e r 2 2 0 5 5 は、I E E E 8 0 2 . 1 D (従来技術 1) もしくは I E E E 8 0 2 . 1 w (従来技術 2) に記載の、R o o t I d e n t i f i e r と等しい情報を格納する領域である。

【 0 1 5 3 】

R o o t P a t h C o s t 2 2 0 5 6 は、I E E E 8 0 2 . 1 D (従来技術 1) もしくは I E E E 8 0 2 . 1 w (従来技術 2) に記載の、R o o t P a t h C o s t と等しい情報を格納する領域である。

【 0 1 5 4 】

B r i d g e I d e n t i f i e r 2 2 0 5 7 は、I E E E 8 0 2 . 1 D (



従来技術 1) もしくは IEEE 8 0 2 . 1 w (従来技術 2) に記載の、B r i d g e I d e n t i f i e r と等しい情報を格納する領域である。

【 0 1 5 5 】

P o r t I d e n t i f i e r 2 2 0 5 8 は、IEEE 8 0 2 . 1 D (従来技術 1) もしくは IEEE 8 0 2 . 1 w (従来技術 2) に記載の、P o r t I d e n t i f i e r と等しい情報を格納する領域である。

【 0 1 5 6 】

M e s s a g e A g e 2 2 0 5 9 は、IEEE 8 0 2 . 1 D (従来技術 1) もしくは IEEE 8 0 2 . 1 w (従来技術 2) に記載の、M e s s a g e A g e と等しい情報を格納する領域である。

【 0 1 5 7 】

M A X A g e 2 2 0 5 A は、IEEE 8 0 2 . 1 D (従来技術 1) もしくは IEEE 8 0 2 . 1 w (従来技術 2) に記載の、M A X A g e と等しい情報を格納する領域である。

【 0 1 5 8 】

H e l l o T i m e 2 2 0 5 B は、IEEE 8 0 2 . 1 D (従来技術 1) もしくは IEEE 8 0 2 . 1 w (従来技術 2) に記載の、H e l l o T i m e と等しい情報を格納する領域である。

【 0 1 5 9 】

F o r w a r d D e l a y 2 2 0 5 C は、IEEE 8 0 2 . 1 D (従来技術 1) もしくは IEEE 8 0 2 . 1 w (従来技術 2) に記載の、F o r w a r d D e l a y と等しい情報を格納する領域である。

【 0 1 6 0 】

図 6 は、IEEE 8 0 2 . 1 D (従来技術 1) および IEEE 8 0 2 . 1 w (従来技術 2) に記載の、T o p o l o g y C h a n g e N o t i f i c a t i o n B P D U フレームの構造を示すフォーマット図である。

【 0 1 6 1 】

M A C D A 2 2 0 1 は、宛先 M A C アドレスを格納する領域である。

【 0 1 6 2 】

MAC SA 2 2 0 2 は、送信元MACアドレスを格納する領域である。

【 0 1 6 3 】

タグ領域 2 2 0 3 は、従来技術には記載されていないが、複数のスパニングツリーを識別する識別子としてのタグを挿入する領域である。前記タグは、VLANタグのほかに、上記特願 2 0 0 2 - 2 0 4 6 7 3 号に開示される拡張タグ、及び他のタグもしくは識別手段の何れか 1 つ以上の組み合わせたタグでもよい。

【 0 1 6 4 】

Type 2 2 0 4 は、フレームのタイプ識別子を格納する領域である。

【 0 1 6 5 】

BPDU領域 2 2 0 5 は、IEEE 8 0 2 . 1 D（従来技術 1）およびIEEE 8 0 2 . 1 w（従来技術 2）に記載の、Topology Change Notification BPDU parametersに相当する情報を格納する領域である。

【 0 1 6 6 】

FCS 2 2 0 6 は、フレームチェックシーケンスを格納する領域である。

【 0 1 6 7 】

GVRPとは、タググループの管理および、現用系と予備系の識別、および、ノード間での各種設定情報の交換のために送受信される、制御フレームのことをいう。

【 0 1 6 8 】

図 5 3 において、拡張タグフレーム 3 3 0 0 および 3 4 0 0 の他のフレームフォーマットについて説明する。なお、以降では図 4 で説明した拡張タグフレーム 3 5 0 0 ~ 3 5 0 8 のフレームフォーマットを拡張タグフレームフォーマット（1）と記し、以降で図 5 3 において説明するフレームフォーマットを拡張タグフレームフォーマット（2）と記すこととする。

【 0 1 6 9 】

図 5 3 の上部はVLANタグ 3 2 0 3 の詳細なフレームフォーマットである。TPID (Tag Protocol Identifier) 2 8 0 0 には、値「0 x 8 1 0 0」が設定されている。なお、標準規格外ではあるが値「0 x 9

100」が使用される場合もある。また、TCI2801はPriorityフィールド2802とCFI2803とVLAN-IDフィールド2804から構成される。

## 【0170】

Priorityフィールド2802にはフレームの優先度が格納され、その優先度の値はIEEE802.1pにおいて規定されている。また、CFIには特殊ルーティング情報の有無またはMACアドレスのフォーマットの種別を示す値が格納され、VLAN-IDフィールド2804にはVLAN-IDが格納される。

## 【0171】

これに対して、図50の下部に示す拡張タグフレームフォーマット(2)では、TPID2800及びTCI2801の中のCFI2803はVLANタグ3203と同様であり、Priorityフィールド2802がPriority/タグTypeフィールド5003に、VLAN-IDフィールド2804が拡張タグ情報フィールド5004に変更される。なお、対応するフィールドのサイズは同様である。

## 【0172】

本拡張タグフレームフォーマット(2)では、Priority/タグTypeフィールド5003において、拡張タグ3500～3508の種別を格納する。拡張タグ3500～2307使用時にもIEEE802.1pをサポートできるように、既存のVLANタグ3203のPriorityフィールド2802(IEEE802.1p)におけるPriority値の一部を拡張タグ3500～3508の種別として用いる。

## 【0173】

具体的には110、100、001、000を拡張タグ3500～3508用として用いることとし、111(予約用)、101(会話型マルチメディア用)、011(クリティカルアプリケーション用)、010(標準的ストリーム用)についてはIEEE802.1p互換とする。

## 【0174】

そのため、使用できる拡張タグ 3 5 0 0 ~ 3 5 0 8 は 4 つに制限しており、例えばフォワーディングタグ 3 5 0 0、ブロードキャストフォワーディングタグ 3 5 0 8、カスタマ分離タグ 3 5 0 1、OAM & P タグ 3 5 0 3 を使用し、P r i o r i t y 値との対応を 0 0 1 = フォワーディングタグ 3 5 0 0、0 0 0 = ブロードキャストフォワーディングタグ 3 5 0 8、1 1 0 = カスタマ分離タグ 3 5 0 1、1 0 0 = OAM & P タグ 3 5 0 3 とする。これにより、上記 4 種類の拡張タグを識別できると共に、I E E E 8 0 2 . 1 p の中の 4 つの優先度をサポートできる。なお、使用する拡張タグの選択とそれに対応する P r i o r i t y 値の設定についてはこの例に限定されるわけではない。

## 【 0 1 7 5 】

また拡張タグフレームフォーマット ( 2 ) では、拡張タグ情報フィールド 5 0 0 4 において、拡張タグ 3 5 0 0 ~ 3 5 0 8 のタグ種別に応じたアドレス情報などの情報が格納される。例えば、フォワーディングタグ 3 5 0 0 の場合、宛先ノードのアドレス情報が格納され、ブロードキャストフォワーディングタグ 3 5 0 8 の場合、送信元ノードのアドレス情報が格納され、カスタマ分離タグ 3 5 0 1 の場合、カスタマの識別情報が格納される。

## 【 0 1 7 6 】

( 第 1 の実施の形態 )

以下、本発明の第 1 の実施の形態について、図面を参照して詳細に説明する。

## 【 0 1 7 7 】

図 7 を参照すると、本発明の第 1 の実施の形態は、ノード 1 1 ~ 1 6、クライアント 9 1 ~ 9 6、リンク 8 1 ~ 8 6、リンク 2 1 ~ 2 8 を含む。

## 【 0 1 7 8 】

ノード 1 1 は、プログラム制御される C P U 等で実現され、以下に挙げる機能を有する。

## 【 0 1 7 9 】

( 1 ) リンク 2 1 もしくはリンク 2 4 より到着したフレームを、リンク 2 4 もしくはリンク 2 1 に転送する。

## 【 0 1 8 0 】

(2) リンク 8 1 より到着したフレームに、転送に必要なタグを付加した上で、リンク 2 1 もしくはリンク 2 4 に転送する。

【 0 1 8 1 】

(3) リンク 2 1 もしくはリンク 2 4 より到着したフレームを、転送に必要なタグを削除した上で、リンク 8 1 に転送する。

【 0 1 8 2 】

(4) スパニングツリーを構成するために、他のノードとの間で制御フレームのを行い、必要に応じてリンクのポートを閉鎖する。

【 0 1 8 3 】

(5) リンクを流れるフレームの流量をモニタする。

【 0 1 8 4 】

ノード 1 2 ～ 1 6 は、ノード 1 1 と同様のノードである。以後、ノード 1 1 ～ 1 6 を代表して、ノード 1 1 を用いて記述を行うが、これらノード 1 1 についての記述は、特に断りのない限り他のノード 1 2 ～ 1 6 においても同様に実現可能である。

【 0 1 8 5 】

クライアント 9 1 は、1 つ以上のクライアントの集合であり、リンク 8 1 を通じて、ノード 1 1 との間でフレームの送受信を行う機能を有する。

【 0 1 8 6 】

クライアント 9 2 ～ 9 6 は、クライアント 9 1 と同様のクライアント群である。以後、クライアント 9 1 ～ 9 6 を代表してクライアント 9 1 を用いて記述を行うが、クライアント 9 1 についての記述は、特に断りのない限り他のクライアント 9 2 ～ 9 6 においても同様に適用可能である。

【 0 1 8 7 】

リンク 8 1 は、クライアント 9 1 からノード 1 1、及び、ノード 1 1 からクライアント 9 1 を結ぶ双方向リンクである。

【 0 1 8 8 】

リンク 8 2 ～ 8 6 は、リンク 8 1 と同様のリンクである。以後、リンク 8 1 ～ 8 6 を代表してリンク 8 1 を用いて記述を行うが、リンク 8 1 についての記述は

、特に断りのない限り他のリンク 8 2 ～ 8 6 においても同様に適用可能である。

#### 【 0 1 8 9 】

リンク 2 1 は、ノード 1 1 からノード 1 2、及び、ノード 1 2 からノード 1 1 を結ぶ、双方向リンクである。

#### 【 0 1 9 0 】

リンク 2 2 ～ 2 6 は、リンク 2 1 と同様のリンクである。以後、リンク 2 2 ～ 2 6 を代表してリンク 2 1 を用いて記述を行うが、リンク 2 1 についての記述は、特に断りのない限り他のリンク 2 2 ～ 2 6 においても同様に適用可能である。

#### 【 0 1 9 1 】

図 8 は、ノード 1 1 の構成を詳細に示した図である。ノード 1 1 は、フレーム転送器 1 1 1 と、タグ挿入器 1 1 2 と、タグ削除器 1 1 3 と、フォーディングテーブル 1 1 4 と、分別器 1 1 5 0 と、ツリーマネージャ 1 1 5 1 と、ツリーマネージャ 1 1 5 2 と、ツリーセクタ 1 1 6 と、タグテーブル 1 1 7 と、設定インタフェース 1 1 8 を含む。

#### 【 0 1 9 2 】

フレーム転送器 1 1 1 は、リンク 2 1 もしくはリンク 2 4、およびタグ挿入器 1 1 2 より受信したフレームを、フォーディングテーブル 1 1 4 の記述に従い、リンク 2 1 もしくはリンク 2 4、および、タグ削除器 1 1 3 もしくはもしくはツリーセクタ 1 1 6 に転送する。

#### 【 0 1 9 3 】

タグ挿入器 1 1 2 は、リンク 8 1 より受信したフレームに対し、タグテーブル 1 1 7 の記述に従いタグを挿入し、フレーム転送器 1 1 1 に転送する。なお、タグテーブル 1 1 7 の記述によっては、タグを挿入せずに、受信したフレームをそのままフレーム転送器 1 1 1 に転送することもできるほか、同一フレームに 0 個以上の複数のタグを挿入し、もしくは、到着したフレームをコピーし、コピーしたそれぞれのフレームに対して、同一もしくは異なるタグを、0 個以上の複数個挿入することも可能である。

#### 【 0 1 9 4 】

タグ削除器 1 1 3 は、フレーム転送器 1 1 1 より受信したフレームに付加され



ているタグを外し、リンク 8 1 に転送する。なお、設定によって、タグを外さずに、受信したフレームをそのままリンク 8 1 に転送することもできる。

#### 【 0 1 9 5 】

フォワーディングテーブル 1 1 4 は、フレーム転送器 1 1 1 からの問い合わせに対し、MAC アドレス、タグ、もしくは入力ポートの他、1 つ以上の組み合わせをキーとして、1 つ以上のフレーム転送先ポートをフレーム転送器 1 1 1 に返答する。キーおよび転送先ポートは、ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 により設定される。

#### 【 0 1 9 6 】

分別器 1 1 5 0 は、受信したフレームのタグに従い出力先ポートを決定し、前記フレームをツリーマネージャ 1 1 5 1、または、ツリーマネージャ 1 1 5 2 に転送する。どのタグが付加されたフレームをツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 に転送するかは、ツリーセクタ 1 1 6 より設定することができる。

#### 【 0 1 9 7 】

ツリーマネージャ 1 1 5 1 は、ツリーセクタ 1 1 6 の指示に従い、スパニングツリーアルゴリズムを利用して、分別器 1 1 5 0 より B P D U を受信し、またフレーム転送器 1 1 1 に対して B P D U を送信し、フォワーディングテーブル 1 1 4 を設定する。さらに、ツリーセクタ 1 1 6 から設定情報を受信し、B P D U のパラメータとして利用する。また、B P D U に含まれる制御情報を抽出し、ツリーセクタ 1 1 6 に通知する。

#### 【 0 1 9 8 】

ツリーマネージャ 1 1 5 2 は、ツリーマネージャ 1 1 5 1 と同様のツリーマネージャである。以後、ツリーマネージャ 1 1 5 1 ~ ツリーマネージャ 1 1 5 2 を代表してツリーマネージャ 1 1 5 1 を用いて記述を行うが、ツリーマネージャ 1 1 5 1 についての記述は、特に断りのない限りツリーマネージャ 1 1 5 2 においても同様に適用可能である。

#### 【 0 1 9 9 】

ツリーセクタ 1 1 6 は、フレーム転送器 1 1 1 より G V R P 等の設定フレー

ムを受信し、ツリーマネージャ 1 1 5 1 もしくは 1 1 5 2 より B P D U に含まれる制御情報を受信し、リソースモニタ 1 1 9 よりリンク情報の通知を受け、または、設定インタフェース 1 1 8 からの設定通知を受け、設定フレームもしくは通知に含まれる情報に沿って、ツリーマネージャ 1 1 5 1、ツリーマネージャ 1 1 5 2、タグテーブル 1 1 7 の設定を行う。また、設定フレームをフレーム転送器 1 1 1 に送信する。

#### 【 0 2 0 0 】

タグテーブル 1 1 7 は、タグ挿入器 1 1 2 からの問い合わせに対し、挿入すべきタグの情報、もしくは、タグを追加しないで転送する命令をタグ挿入器 1 1 2 に返答する。挿入するタグもしくは、タグ挿入なしで転送命令は、ツリーセクタ 1 1 6 より設定される。同一フレームに 0 個以上の複数のタグを挿入する設定や、到着したフレームをコピーし、コピーしたそれぞれのフレームに対して、同一もしくは異なるタグを、0 個以上の複数個挿入する設定も可能である。

#### 【 0 2 0 1 】

設定インタフェース 1 1 8 は、シリアル接続もしくは T E L N E T 等のコマンドラインインタフェース、もしくは W E B サーバ等を通じて、ユーザからのツリー選択命令や、ノード削除要求、リンクコスト、スパニングツリーパラメータ値等を、ツリーセクタ 1 1 6 に伝達する。

#### 【 0 2 0 2 】

リソースモニタ 1 1 9 は、ノードの各リンクポートの状況を監視し、リンクの接続を検出した場合には、ツリーセクタ 1 1 6 にリンクアップ通知を送信する。また、リンクを通過するフレームの累積バイト数、通過 TCP セッション数、HTTP リクエスト数のうち 1 つ以上の値をカウントして保持し、ツリーセクタ 1 1 6 の要求により、保持している値をツリーセクタ 1 1 6 に通知するほか、ツリーセクタ 1 1 6 からの命令により、保持している値を 0 にリセットする。さらに、ツリーセクタ 1 1 6 よりあらかじめ指定された種類のフレーム通過を監視し、前記監視対象フレームが通過すると、ツリーセクタ 1 1 6 に通知する。

#### 【 0 2 0 3 】

図 9 は、本実施の形態の図 8 における、タグをキーとして出力ポートを決定す

る、フォワーディングテーブル 1 1 4 の構成例である。

#### 【 0 2 0 4 】

タグフィールド 1 1 4 1 は、サーチのインデックスになるフィールドであり、このフィールドの情報と、受信したフレームのタグに書かれている内容が一致するかどうかを調べる。

#### 【 0 2 0 5 】

出力ポート 1 1 4 2 は、受信したフレームのタグに書かれている内容とタグにフィールド 1 1 4 1 の内容が一致した場合に、前記フレームをどのポートに転送すべきか記述するフィールドである。

#### 【 0 2 0 6 】

なお、本実施の形態は、本動作例に示すような、タグの内容によって転送先ポートを決定するタグフォワーディングを行う場合のみならず、従来より行われている、MAC アドレスによって転送先を決定する通常の MAC アドレス転送においても、同様に適用可能である。この場合は、出力ポートフィールド 1 1 4 2 に、複数のポートが記載される。

#### 【 0 2 0 7 】

図 1 0 は、本発明の第 1 の実施の形態の図 8 における、ツリーマネージャ 1 1 5 1 の構成を詳細に示した図である。ツリーマネージャ 1 1 5 1 は、タグ削除器 1 1 5 1 1 と、BPDU 送受信器 1 1 5 1 2 と、タグ挿入器 1 1 5 1 3 と、ツリーコントローラ 1 1 5 1 4 と、ツリーテーブル 1 1 5 1 5 含む。

#### 【 0 2 0 8 】

タグ削除器 1 1 5 1 1 は、分別器 1 1 5 0 より入力されたフレームに挿入されているタグを削除し、BPDU 送受信機 1 1 5 1 2 に転送する。もし分別器 1 1 5 0 より受信したフレームにタグが添付されていない場合には、受信したフレームをそのまま BPDU 送受信機 1 1 5 1 2 に転送する。

#### 【 0 2 0 9 】

BPDU 送受信器 1 1 5 1 2 は、タグ削除器 1 1 5 1 1 より BPDU を受信し、フレームに含まれる情報を、BPDU 受信通知によりツリーコントローラ 1 1 5 1 4 に通知する。また、ツリーコントローラ 1 1 5 1 4 より BPDU 送信通知

を受け、フレームを生成してタグ挿入器 1 1 5 1 3 に送信する。

【 0 2 1 0 】

タグ挿入器 1 1 5 1 3 は、B P D U 送受信機 1 1 5 1 2 よりフレームを受信し、あらかじめ設定されているタグを挿入し、フレーム転送器 1 1 1 に送信する。なお、タグを挿入せずに、フレームをそのまま転送する設定も可能である。

【 0 2 1 1 】

ツリーコントローラ 1 1 5 1 4 は、以下に示す 4 つの機能を有する。

【 0 2 1 2 】

(1) 停止動作 (初期状態) : ツリーセクタ 1 1 6 からの停止命令にしたがい、B P D U 送受信機 1 1 5 1 2 への B P D U 送信通知を停止する。また、ツリーテーブル 1 1 5 1 5 に対して、すべてのリンクが down しているとして、ポートの状態を登録する。

【 0 2 1 3 】

(2) 開始動作 : ツリーセクタ 1 1 6 からの開始命令にしたがい、B P D U 送受信機 1 1 5 1 2 への B P D U 送信通知を開始する。また、開始命令に含まれる情報を元に、ツリーテーブル 1 1 5 1 5 に対して、アップ状態のポートを登録する。

【 0 2 1 4 】

(3) B P D U 受信動作 : B P D U 送受信機 1 1 5 1 2 からの B P D U 受信通知を受け取り、ツリーテーブル 1 1 5 1 5 を更新する。また、B P D U 受信通知に含まれる現用系ツリーおよび予備系ツリーの識別情報を抽出し、ツリーセクタ 1 1 6 内のメインコントローラ 1 1 6 4 に通知する。

【 0 2 1 5 】

(4) トポロジ更新動作 : 前記停止動作、前記開始動作、および前記 B P D U 受信動作後に、従来技術 1 もしくは従来技術 2 に示すスパニングツリープロトコルに従い、ツリーテーブル 1 1 5 1 5 を参照し、必要であれば、ツリーテーブル 1 1 5 1 5 の設定、フォワーディングテーブル 1 1 4 の設定および、B P D U の送信を行う。送信する B P D U には、現用系ツリーおよび予備系ツリーの識別情報も含まれる。また、ツリーセクタ 1 1 6 に対し、ツリーの再計算の結果、ト

ポロジに変更が生じるか生じないかを通知する動作を行う。

#### 【 0 2 1 6 】

ツリーテーブル 1 1 5 1 5 は、従来技術 1 もしくは従来技術 2 に示すスパニングツリープロトコルに必要な、ポートの状態およびノードの状態に関するパラメータが記述されているテーブルである。各ポートもしくはリンクの優先度並びにリンクコストも、このテーブルに記述されている。第 1 の実施の形態では、リンクコストとして、リンク帯域の太さを用いる場合を想定して説明する。

#### 【 0 2 1 7 】

図 1 1 は、本発明の第 1 の実施の形態の図 8 における、ツリーセクタ 1 1 6 の構成を詳細に示した図である。ツリーセクタ 1 1 6 は、タグ削除器 1 1 6 1 と、GVRP 送受信器 1 1 6 2 と、タグ挿入器 1 1 6 3 と、メインコントローラ 1 1 6 4 と、安定タイマ 1 1 6 5 と、到着間隔タイマ 1 1 6 6 とを含む。

#### 【 0 2 1 8 】

タグ削除器 1 1 6 1 は、フレーム転送器 1 1 1 より入力されたフレームに挿入されているタグを削除し、GVRP 送受信機 1 1 6 2 に転送する。もしフレーム転送器 1 1 1 より受信したフレームにタグが添付されていない場合には、受信したフレームをそのまま GVRP 送受信機 1 1 6 2 に転送する。

#### 【 0 2 1 9 】

GVRP 送受信器 1 1 6 2 は、タグ削除器 1 1 6 1 より制御フレームを受信し、フレームに含まれる情報を、GVRP フレーム受信通知によりメインコントローラ 1 1 6 4 に通知する。また、メインコントローラ 1 1 6 4 より GVRP フレーム送信通知を受け、フレームを生成してタグ挿入器 1 1 6 3 に送信する。

#### 【 0 2 2 0 】

タグ挿入器 1 1 6 3 は、GVRP 送受信機 1 1 6 2 よりフレームを受信し、あらかじめ設定されているタグを挿入し、フレーム転送器 1 1 1 に送信する。なお、タグを挿入せずに、フレームをそのまま転送する設定も可能である。

#### 【 0 2 2 1 】

メインコントローラ 1 1 6 4 は、以下に示す 4 つの機能を有する。

#### 【 0 2 2 2 】

(1) リンクアップ検出：メインコントローラ 1 1 6 4 は、リソースモニタ 1 1 9 よりリンクアップ通知を受け、現在予備系となっているツリーマネージャに対して、リンクアップ（開始命令）を通知する。開始命令には、アップしているリンクの情報が格納される。また、開始命令送信後に、安定タイマ 1 1 6 5 をセットする。安定タイマ 1 1 6 5 の満了通知を受信すると、タグテーブル 1 1 7 に対して挿入タグの変更命令を通知し、さらに、予備系と現用系の登録を逆転させるよう、GVRP送受信機 1 1 6 2 に対して、新ツリーのルートノードに対してツリー切り替え要求フレームを送信するよう命令する。さらに、安定タイマ 1 1 6 5 をセットし、タイマ満了後に自ノードを旧ツリーに追加する。

## 【 0 2 2 3 】

(2) ノード削除要求受信：設定インタフェース 1 1 8 よりノード削除要求通知を受信すると、現在予備系となっているツリーマネージャに対して停止命令を送信する。また停止命令の送信後に、安定タイマ 1 1 6 5 をセットし、安定タイマ 1 1 6 5 の満了通知を受信すると、タグテーブル 1 1 7 に対して挿入タグの変更命令を通知し、さらに、予備系と現用系の登録を逆転させるよう、GVRP送受信機 1 1 6 2 に対して、新ツリーのルートノードに対して利用タググループ変更GVRPフレームを送信するよう命令する。さらに、安定タイマ 1 1 6 5 をセットし、タイマ満了後に設定インタフェース 1 1 8 に対して、自ノード削除許可の表示を行う。

## 【 0 2 2 4 】

(3) 利用タググループ変更GVRP受信：自ノードが新ツリーのルートノードである場合、利用タググループ変更GVRPフレームを受信すると、自ノードから送信するBPDUに、現用系フラグを付加して送信するよう、ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 に対して命令する。さらに、GVRP送受信機 1 1 6 2 に対して、旧ツリーのルートノードに対して、現用系フラグの添付を取り消すよう命令する。

## 【 0 2 2 5 】

(4) 現用系ビット変更通知受信：ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 は、BPDUの受信時にBPDUに付加された現用系フラグ



を確認し、自グループが現用系であるか予備系であるか確認し、メインコントローラに通知する。通知を受けた結果、もし、現用系と予備系に変更がある場合には、タグテーブル 1 1 7 に対して挿入タグ変更通知を送信して、現用系と予備系の登録を逆転させる。

## 【 0 2 2 6 】

(5) 指定フレームの通過通知受信：リソースモニタ 1 1 9 より、あらかじめ設定した監視対象フレームが通過したことの通知を受信すると、到着間隔タイマ 1 1 6 6 に対してセット通知を送信する。到着間隔タイマ 1 1 6 6 から、タイマ満了通知が到着した場合は、監視対象フレームの到着間隔がセット通知によって設定した時間より長いことがわかる。これにより B P D U の到着間隔が長くなったことや、予備系を流れるフレームの到着間隔が長くなったことなどを検出できる。

## 【 0 2 2 7 】

安定タイマ 1 1 6 5 は、メインコントローラ 1 1 6 4 より送信されたセット通知の受信から、あらかじめ設定された時間が経過すると、メインコントローラ 1 1 6 4 に対してタイマ満了通知を送信する。

## 【 0 2 2 8 】

到着間隔タイマ 1 1 6 6 は、メインコントローラ 1 1 6 4 からセット命令を受信すると、現在保持している時間を 0 にリセットしてタイマを作動させ、セット命令によって指定された時間が経過すると、メインコントローラ 1 1 6 4 に対してタイマ満了通知を送信する。

## 【 0 2 2 9 】

図 1 2 は、本発明の第 1 の実施の形態の図 1 1 における、メインコントローラ 1 1 6 4 の状態遷移を詳細に示した流れ図である。

## 【 0 2 3 0 】

以降、2 系統存在するスパニングツリーのうち、新しくネットワークに侵入するデータフレームの転送に利用しているスパニングツリーを現用ツリーもしくは現用系ツリー、現用ツリーとなっていないツリーを予備用ツリーもしくは予備系ツリーと表現する。

## 【 0 2 3 1 】

また、現用系ツリーの生成を行うツリーマネージャを現用系ツリーマネージャ、予備系ツリーの生成を行うツリーマネージャを、予備系ツリーマネージャと呼ぶ。

## 【 0 2 3 2 】

状態 1 1 6 4 1 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 1 であるか、ツリーマネージャ 1 1 5 2 であるか判別できていない状態であり、かつ、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 および、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能は無効にされ、B P D U 受信機能のみが有効になっている状態である。

## 【 0 2 3 3 】

状態 1 1 6 4 2 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 1 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 2 であり、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能が無効にされ、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能も無効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくは無効に関わらず、常に有効である。

## 【 0 2 3 4 】

状態 1 1 6 4 3 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 1 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 2 であり、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能が無効にされ、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能は有効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくは無効に関わらず、常に有効である。

## 【 0 2 3 5 】

状態 1 1 6 4 4 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 2 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 1 であり、ツリーマネ

ージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能が有効にされ、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能は無効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくは無効に関わらず、常に有効である。

## 【 0 2 3 6 】

状態 1 1 6 4 5 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 2 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 1 であり、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能が有効にされ、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能も有効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくは無効に関わらず、常に有効である。

## 【 0 2 3 7 】

状態 1 1 6 4 6 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 1 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 2 であり、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能が有効にされ、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能も有効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくは無効に関わらず、常に有効である。

## 【 0 2 3 8 】

状態 1 1 6 4 7 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 1 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 2 であり、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能が有効にされ、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能が無効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくは無効に関わらず、常に有効である。

## 【 0 2 3 9 】

状態 1 1 6 4 8 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 2 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 1 であり、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能が無効にされ、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能が有効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくはは無効に関わらず、常に有効である。

## 【 0 2 4 0 】

状態 1 1 6 4 9 は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 2 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 1 であり、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能が無効にされ、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能が無効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくはは無効に関わらず、常に有効である。

## 【 0 2 4 1 】

次に、図 1 2 を参照して、メインコントローラ 1 1 6 4 の動作について説明する。

## 【 0 2 4 2 】

メインコントローラ 1 1 6 4 は、リソースモニタ 1 1 9 より新たにネットワークに接続されたことの通知を受信すると、ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 からの現用系通知の到着を待つ。ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 より B P D U に含まれる現用系通知を受信すると、前記通知において指定されたツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 を現用、予備用に指定されたツリーマネージャ 1 1 5 1 もしくは 1 1 5 2 を予備用に設定し、状態 1 1 6 4 2 もしくは状態 1 1 6 4 9 に遷移する。ここでは、状態 1 1 6 4 2 に遷移した場合を例に解説するが、以後の解説は状態 1 1 6 4 9 に遷移した場合においても同様である。（状態 1 1 6 4 1）

## 【 0 2 4 3 】

メインコントローラ 1 1 6 4 は、ツリーマネージャ 1 1 5 1 を現用、ツリーマネージャ 1 1 5 2 を予備用に設定する。さらに、前記各ツリーマネージャ 1 1 5 1 および 1 1 5 2 に対して、B P D U 送信停止命令を出す。（状態 1 1 6 4 2）

## 【 0 2 4 4 】

メインコントローラ 1 1 6 4 は、状態 1 1 6 4 2 において、もし設定インタフェース 1 1 8 よりノードの追加要求を受信した場合は、状態 1 1 6 4 3 に遷移する。また、もしツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 より現用系通知を受信し、現用系と予備系の関係に変更があった場合には、状態 1 1 6 4 9 に遷移する。（状態 1 1 6 4 2）

## 【 0 2 4 5 】

メインコントローラ 1 1 6 4 は、ツリーマネージャ 1 1 5 2 に対してリンク u p の通知を行い、同時に、ツリーマネージャ 1 1 5 2 に対して B P D U の送信許可を行う。さらに、安定タイマ 1 1 6 5 を作動させる。（状態 1 1 6 4 3）

## 【 0 2 4 6 】

メインコントローラ 1 1 6 4 は、安定タイマ 1 1 6 5 よりタイマ満了通知を受信すると、現用に登録されているツリーマネージャ 1 1 5 1 と、予備用に登録されているツリーマネージャ 1 1 5 2 を入れ替え、新しく現用をツリーマネージャ 1 1 5 2、予備用をツリーマネージャ 1 1 5 1 とする。さらに、G V R P 送受信機 1 1 6 2 を通じて、新ツリーのルートノードに対して、利用タググループ変更通知を送信する。この利用タググループ変更通知の内容は、B P D U に反映され、全ノードに伝達される。その後、安定タイマ 1 1 6 5 を作動させる。（状態 1 1 6 4 4）

## 【 0 2 4 7 】

メインコントローラ 1 1 6 4 は、安定タイマ 1 1 6 5 よりタイマ満了通知を受信すると、ツリーマネージャ 1 1 5 1 に対してリンク u p の通知を行い、同時に、ツリーマネージャ 1 1 5 2 に対して B P D U の送信許可を行う。通常は、この状態で安定する。（状態 1 1 6 4 5）

## 【 0 2 4 8 】

メインコントローラ 1 1 6 4 は、状態 1 1 6 4 5 において、ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 より B P D U に含まれる現用系通知を受信し、もし現用系と予備系の関係に変更が生じた場合には、状態 1 1 6 4 6 に遷移する。そして、現用に登録されているツリーマネージャ 1 1 5 2 と、予備用に登録されているツリーマネージャ 1 1 5 1 を入れ替え、新しく現用をツリーマネージャ 1 1 5 1、予備用をツリーマネージャ 1 1 5 2 とする。（状態 1 1 6 4 5）

【 0 2 4 9 】

メインコントローラ 1 1 6 4 は、状態 1 1 6 4 5 において、もし設定インタフェース 1 1 8 より、ノード削除要求を受信した場合は、状態 1 1 6 4 4 に遷移する。（状態 1 1 6 4 5）

【 0 2 5 0 】

メインコントローラ 1 1 6 4 は、ツリーマネージャ 1 1 5 1 に対して、接続されている全リンクリンク d o w n の通知を行い、同時に、ツリーマネージャ 1 1 5 1 に対して B P D U の送信停止命令を出す。さらに、安定タイマ 1 1 6 5 を作動させる。（状態 1 1 6 4 4）

【 0 2 5 1 】

メインコントローラ 1 1 6 4 は、安定タイマ 1 1 6 5 よりタイマ満了通知を受信すると、現用に登録されているツリーマネージャ 1 1 5 2 と、予備用に登録されているツリーマネージャ 1 1 5 1 を入れ替え、新しく現用をツリーマネージャ 1 1 5 1、予備用をツリーマネージャ 1 1 5 2 とする。さらに、G V R P 送受信機 1 1 6 2 を通じて、新ツリーのルートノードに対して、利用タググループ変更通知を送信する。この、利用タググループ変更通知の内容は、B P D U に反映され、全ノードに伝達される。その後、安定タイマ 1 1 6 5 を作動させる。（状態 1 1 6 4 3）

【 0 2 5 2 】

メインコントローラ 1 1 6 4 は、安定タイマ 1 1 6 5 よりタイマ満了通知を受信すると、ツリーマネージャ 1 1 5 1 に対してリンク d o w n の通知を行い、同時に、ツリーマネージャ 1 1 5 1 に対して B P D U の受信停止命令を出す。さら



に、無条件で状態 1 1 6 4 1 に遷移し、ノード切離しまで待機する。（状態 1 1 6 4 2）

図 1 3 は、本実施の形態の図 8 における、宛先 M A C アドレスをキーとして挿入するタグを決定する、タグテーブル 1 1 7 の構成例である。

#### 【 0 2 5 3 】

宛先 M A C アドレス 1 1 7 1 は、サーチのインデックスになるフィールドであり、このフィールドの情報と、受信したフレームの宛先 M A C アドレスフィールド、すなわち、M A C D A フィールドに書かれている内容が一致するかどうか調べ、一致した場合には、前記受信フレームに対して、挿入タグフィールド 1 1 7 2 に記載のタグを挿入する。

#### 【 0 2 5 4 】

挿入タグフィールド 1 1 7 2 は、宛先 M A C アドレスフィールド 1 1 7 1 に対する、挿入すべきタグが記載されるフィールドである。本実施の形態では、現在において現用系となっているタググループのタグが挿入される。この挿入タグフィールド 1 1 7 2 は、ツリーセクタ 1 1 6 によって、現在現用系となっているタグに書き換えられる。

#### 【 0 2 5 5 】

図 8、図 1 4、図 1 5 及び図 1 6 を参照して、本実施の形態において、ノード 1 7 を追加する場合の動作について、具体例を用いて詳細に説明する。

#### 【 0 2 5 6 】

初期状態（ノード 1 7 がリンク 2 9、3 0 に接続される前の状態）における、2 系統のスパニングツリーは、同じ接続関係となる。これは、スパニングツリーを設定するプロトコルは、ノード、リンクのプライオリティ値等の情報に基づいてスパニングツリーを設定するため、同一のネットワークについてスパニングツリーを 2 つ設定すると結果として同じ接続関係となるものである。この初期状態で、何れか一方のスパニングツリーを現用、他方を予備として設定し（具体的には、ツリーマネージャ 1 1 5 1、1 1 5 2 がツリーテーブル 1 1 5 1 5 を書き換え、ツリーセクタ 1 1 6 がタグテーブル 1 1 7 を書き換えることで各設定を行う）、このネットワークは現用のスパニングツリーを用いて運用される。

【 0 2 5 7 】

図 1 0 に示すネットワークにおいて、初期状態で、太線で示すスパニングツリー 5 1 が 2 つ設定されている場合を想定する。

【 0 2 5 8 】

図 1 4 を参照すると、本動作例では、ノード 1 1 ～ 1 7、リンク 2 1 ～ 3 0、およびツリー 5 1 を有する。ただし、ノード 1 7、リンク 2 9 およびリンク 3 0 は、初期状態では接続されていない。

【 0 2 5 9 】

また、図 1 5 は、本動作例において、ノード 1 7 が追加された後のスパニングツリー 5 2 の状態を示す。ツリー 5 2 は図 1 1 において太線で示されている。

【 0 2 6 0 】

ノード 1 1 ～ 1 6 の全ノード、全ポートが所属するタググループが 2 つすでに設定されており、1 つめのタググループをタググループ 4 1、2 つめのタググループをタググループ 4 2 と呼ぶ。

【 0 2 6 1 】

なお、基本的には全ノード、全ポートを 2 つのタググループに加入させるが、一部のポートもしくはノードのみで構成させるタググループを作成しても良い。以降、全ノード、全ポートが 2 つのタググループに加入するものとして説明する。

【 0 2 6 2 】

ノード 1 1 ～ 1 6 には、独立して動作するスパニングツリー回路が 2 つ存在し、タググループ 4 1 上で動作するスパニングツリーをツリー 5 1、タググループ 4 2 上で動作するスパニングツリーをツリー 5 2 と呼ぶ。

【 0 2 6 3 】

スパニングツリーは必ず 2 系統作成するが、タググループは必ずしも 2 つ作成する必要は無い。前記タググループ 4 1 のみ設定してタググループ 4 2 を利用せず、タググループ 4 1 上で動作するスパニングツリーをツリー 5 1、タググループに属さないで動作するスパニングツリーをツリー 5 2 と呼ぶこともできる。また逆に、前記タググループ 4 2 のみ設定してタググループ 4 1 を利用せず、タグ

グループに属さないで動作するスパニングツリーをツリー 5 1、タググループ 4 2 上で動作するスパニングツリーをツリー 5 2 と呼ぶこともできる。

#### 【 0 2 6 4 】

ここでは、特にタググループ 4 1 とタググループ 4 2 の両方を用いた場合について説明を行うが、タググループ 4 1 及びタググループ 4 2 の両方を用いた場合の動作は、タググループ 4 1 のみ、もしくはタググループ 4 2 のみを用いた場合においても、同様に適用可能である。

#### 【 0 2 6 5 】

このネットワークでは、設定インタフェース 1 1 8 からの初期設定によって、ツリー 5 1 が現用系のスパニングツリーとなり、ツリー 5 2 が予備系のスパニングツリーとなるように設定されており、ツリー 5 1 の B P D U には現用系フラグおよびタググループ 4 1 のタグが付き、ツリー 5 2 の B P D U には予備系フラグおよびタググループ 4 2 のタグが付く。

#### 【 0 2 6 6 】

全ノードが、現用系フラグ又は予備系フラグが付された B P D U フレームを IEEE Std 802.1D や IEEE Std 802.1w で規定される一定の周期で互いに送信し合うことにより、ツリー 5 1 である現用系のスパニングツリー又はツリー 5 2 である予備系のスパニングツリーが構築される。

#### 【 0 2 6 7 】

上記現用系フラグ又は予備系フラグは、図 5 に示した B P D U フレームのフィールドのうち、例えばタグ領域 2 2 0 3、Type 2 2 0 4、B P D U Type 2 2 0 5 3 等のフィールドを使用することによって表すことができる。

#### 【 0 2 6 8 】

現在、ネットワークの開始から十分に時間が経過し、ツリー 5 1 およびツリー 5 2 の各ツリーは、タググループ 4 1 およびタググループ 4 2 のタグが付加された B P D U フレームの交換が十分に行われた結果、共にノード 1 1 をルートノードとして安定しているとする。

#### 【 0 2 6 9 】

安定とは、スパニングツリーのツリー構造が、十分に長い時間変化しない状態

になっている状態のことを言う。

【 0 2 7 0 】

ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U にはタググループ 4 2 のタグが付く。具体的には、ツリー 5 1 を用いて伝送される B P D U のタグ領域 2 2 0 3 には、この B P D U がタググループ 4 1 に属することを示す値が記載され、ツリー 5 2 を用いて伝送される B P D U のタグ領域 2 2 0 3 には、この B P D U がタググループ 4 2 に属することを示す値が記載される。

【 0 2 7 1 】

また、現時点では、ツリー 5 1 が現用系として設定されているので、クライアントからノード 1 1 ～ 1 6 に送信されたデータには、タグ挿入器 1 1 2 によって、タググループ 4 1 のタグが付加されている。具体的には、データ信号のタグ領域に、タググループ 4 1 に属することを示す値が記載される。そして、このタグが付加されたデータは、フレーム転送器 1 1 1 によって、現時点で、現用系として設定されているツリー 5 1 に沿って転送されている。

【 0 2 7 2 】

ノード 1 7 は、リンク 2 9 およびリンク 3 0 と接続されると、どのタググループにも加入せず、B P D U の受信を開始する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 1 となる。）

【 0 2 7 3 】

ノード 1 7 は、各タググループの B P D U を受信すると、現時点での現用系がタググループ 4 1 で、現時点での予備系がタググループ 4 2 であると確認する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 2 である。）そして、自ノードをタググループ 4 2 にのみ参加させて B P D U の送受信を行い、タググループ 4 1 では B P D U は受信のみ行い、送信を行わないように設定する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 3 である。）

【 0 2 7 4 】

ノード 1 7 が追加されると、タググループ 4 2 のメンバに変更が発生するため

、スパニングツリープロトコルにより、ツリー 5 2 を更新する動作が開始される。すなわち、ノード 1 7 が B P D U を送信して、それを隣接ノードが受信すると、隣接ノードはトポロジ状況に変更があったと認識し、ツリー 5 2 の更新動作を開始する。タググループ 4 1 のメンバには変更が無いため、ツリー 5 1 は更新されない。

## 【 0 2 7 5 】

クライアントから送信されるフレームには、各ノード 1 1 ～ 1 6 によって、今までどおりタググループ 4 1 のタグが付加され、ツリー 5 1 に沿って転送され続ける。

## 【 0 2 7 6 】

ここで、段落 0 1 3 5 に記載したツリー 5 2 の更新動作によって、ツリー 5 2 がルートノードをノード 1 1 として、安定したとする。このときのツリー 5 2 の構成を図 1 5 に示す。

## 【 0 2 7 7 】

ノード 1 7 は、ネットワークに接続されてから一定時間経過後に、ツリー 5 2 が安定したと判断し、利用タググループ変更通知を、ツリー 5 2 のルートノードである、ノード 1 1 に送信し、ツリー 5 2 を予備系から現用系に遷移させるよう命令する。この命令には、例えば制御フレーム（G V R P）を利用する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 4 である。）

## 【 0 2 7 8 】

なお、ツリー 5 2 が安定したことの検出は、ノード 1 7 がネットワークに接続されてから一定時間経過することで検出するほか、ノード 1 7 におけるツリー 5 2 の B P D U 到着間隔が一定時間以上となることで検出することもできる。

## 【 0 2 7 9 】

利用タググループ変更通知を受信したノード 1 1 は、ツリー 5 1 のルートノードであるノード 1 1 に対して利用タググループ変更通知を送信し、ツリー 5 1 を予備系に遷移させるよう命令する。この利用タググループ変更通知には、例えば制御フレーム（G V R P）を利用する。そして、タググループ 4 2 のタグを付け

、ツリー 5 2 のために送信する B P D U に、現用系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

#### 【 0 2 8 0 】

ツリー 5 1 のノード 1 1 は、利用タググループ変更通知を受信し、ツリー 5 1 を予備系に遷移させ、タググループ 4 1 のフラグを付け、ツリー 5 1 のために送信する B P D U に、予備系フラグを立てる。この予備系フラグは、例えば、図 1 の B P D U のうちの予め決めてあるフィールドに設定される。設定は、ツリーマネージャ 1 1 5 1、1 1 5 2 がツリーテーブル 1 1 5 1 5 を書き換え、ツリーセクタ 1 1 6 がタグテーブル 1 1 7 を書き換えることで実行される。B P D U は、各ノードで転送されながら、全ノードに伝播する。

#### 【 0 2 8 1 】

ノード 1 1 ～ 1 7 は、タググループ 4 2 のタグが付いた B P D U に現用系フラグが付加させたのを確認し、クライアントより転送されてくるフレームに付加するタグを、タググループ 4 1 からタググループ 4 2 に切り替える。このとき、タグテーブル 1 1 7 内の、挿入タグフィールド 1 1 7 2 を書き換えられる。このタグが付加されたフレームは、ツリー 5 2 に沿って転送される。

#### 【 0 2 8 2 】

以上の切り替えが完了してしばらく経つと、ツリー 5 1 を流れるフレームはなくなる。

#### 【 0 2 8 3 】

ノード 1 7 は、利用タググループ変更通知の送信から一定時間経過後に、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、自ノードをタググループ 4 1 に加入させ、次のトポロジ変更にも備える。自ノードをタググループ 4 1 に加入させるために、ノード 1 7 は、ツリーマネージャ 1 1 5 1 内のツリーコントローラに対して、B P D U 送信を許可し、B P D U 送受信機 1 1 5 1 2 からの B P D U 送信を許可する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 5 となる。）

このとき、スパニングツリー 5 1 について、再構成が行われるので、スパニングツリー 5 1 に着目してネットワークを見ると、ネットワークは停止する。しか



し、この間もネットワークにおける通信は、スパニングツリー 5 2 を用いて行われているので、ノード 1 7 の追加に伴って、輻輳、フレームの到着遅延等の問題は発生しない。

【 0 2 8 4 】

なお、ノード 1 7 をタググループ 4 1 に加入させる動作は、ツリー 5 2 のルートノードであるノード 1 1、もしくは、ツリー 5 1 のルートノードであるノード 1 1 が行っても良い。

【 0 2 8 5 】

ツリー 5 2 のルートノードであるノード 1 1 が、ノード 1 7 をタググループ 4 1 に加入させる場合は、ノード 1 1 がノード 1 7 から利用タググループ変更通知を受信してから一定時間経過後に、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、ノード 1 7 に対して G V R P フレームを送信し、タググループ 4 1 に加入するよう命令する。

【 0 2 8 6 】

ツリー 5 1 のルートノードであるノード 1 1 が、ノード 1 7 をタググループ 4 1 に加入させる場合は、ノード 1 1 がノード 1 1 から利用タググループ変更通知を受信してから一定時間経過後に、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、ノード 1 7 に対して G V R P フレームを送信し、タググループ 4 1 に加入するよう命令する。この場合は、ツリー 5 1 のノード 1 1 は、直接（ツリー 5 2 のノード 1 1 を介さずに）、ノード 1 7 に対して G V R P フレームを送信する。ノード 1 7 をタググループ 4 1 に加入させる動作では、B P D U に挿入するフラグに変更が生じないので、ルートノードを介す必要はないからである。

【 0 2 8 7 】

以上のようにして、ネットワークを停止させることなく、ノード 1 7 を追加することができた。以降ノードの追加の際は、同様の動作を繰り返す。ただし、上記説明文中のタググループ 4 1 とタググループ 4 2 は適宜入れ替わる。

【 0 2 8 8 】

図 1 6 は、以上に述べたノード 1 7 の追加動作を表すシーケンス図である。

【 0 2 8 9 】

矢印 3 1 は、タググループ 4 1 を示すタグが挿入された、現用系フラグ付き B P D U の流れを示す。

【 0 2 9 0 】

矢印 3 2 は、タググループ 4 2 を示すタグが挿入された、予備系フラグ付き B P D U の流れを示す。

【 0 2 9 1 】

矢印 3 3 は、タググループ 4 2 を示すタグが挿入された、現用系フラグ付き B P D U の流れを示す。

【 0 2 9 2 】

矢印 3 4 は、タググループ 4 1 を示すタグが挿入された、予備系フラグ付き B P D U の流れを示す。

【 0 2 9 3 】

矢印 3 5 は、タググループを示すタグが挿入されていない、G V R P フレーム等による利用タググループ変更通知の流れを示す。

【 0 2 9 4 】

次に、図 1 5 および図 1 4 を参照して、本実施の形態において、ノード 1 7 を削除する場合の動作について、具体例を用いて詳細に説明する。

【 0 2 9 5 】

図 1 5 および図 1 4 を参照すると、本動作例では、ノード 1 1 ～ 1 7、およびリンク 2 1 ～ 3 0 を有する。

【 0 2 9 6 】

ノード 1 1 ～ 1 7 の全ポートが所属するタググループが 2 つすでに設定されており、1 つめのタググループをタググループ 4 1、2 つめのタググループをタググループ 4 2 と呼ぶ。

【 0 2 9 7 】

ノード 1 1 ～ 1 7 には、独立して動作するスパニングツリー経路が 2 つ存在し、タググループ 4 1 上で動作するスパニングツリーをツリー 5 1、タググループ 4 2 上で動作するスパニングツリーをツリー 5 2 と呼ぶ。

## 【 0 2 9 8 】

図 1 5 では、ツリー 5 2 は太線で表され、ノード 1 1 をルートノードとして安定しているとする。

## 【 0 2 9 9 】

ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U にはタググループ 4 2 のタグがつく。

## 【 0 3 0 0 】

現在、クライアントからノード 1 1 ～ 1 7 に送信されたデータには、ツリー 5 2 が現用系であるので、タググループ 4 2 のタグが付加されている。このタグが付加されたデータは、ツリー 5 2 に沿って転送されている。

## 【 0 3 0 1 】

ノード 1 7 は、各タググループの B P D U を受信しており、現時点での現用系がタググループ 4 2 で、現時点での予備系がタググループ 4 1 であるとすでに確認しているとする。

## 【 0 3 0 2 】

ノード 1 7 は、設定インタフェースもしくはその他の手段により削除要求を受けると、自ノード 1 7 を現用系であるタググループ 4 2 にのみ参加させ、タググループ 4 1 には参加させないように設定する。このとき、ノード 1 7 は、タググループ 4 1 の B P D U 送信を停止する。

## 【 0 3 0 3 】

この設定により、ノード 1 7 の隣接ノードにおいて B P D U を受信しなくなることでノード 1 7 が削除されたことが認識され、タググループ 4 1 のメンバに変更が発生するため、ツリー 5 1 を更新する動作が開始される。タググループ 4 2 のメンバには変更が無いため、ツリー 5 2 は更新されない。

## 【 0 3 0 4 】

クライアントからのフレームは、ノード 1 1 ～ ノード 1 7 で、今までどおりタググループ 4 2 のタグが付加され、ツリー 5 2 に沿って転送され続ける。

## 【 0 3 0 5 】

ここで、ツリー 5 1 はルートノードをノード 1 1 として、ノード 1 7 を参加さ

せないで安定した状態を示している。ツリー 5 1 の構成を図 1 4 に示す。

【 0 3 0 6 】

なお、ここでいう安定とは、スパニングツリーのツリー構造が、十分に長い時間変化しない状態になっている状態のことを言う。

【 0 3 0 7 】

ノード 1 7 は、タググループ 4 1 不参加の設定から一定時間経過後に、ツリー 5 1 が安定したと判断し、利用タググループ変更通知を、ツリー 5 1 のルートノードである、ノード 1 1 に送信し、ツリー 5 1 を予備系から現用系に遷移させるよう命令する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 4 である。）

【 0 3 0 8 】

利用タググループ変更通知を受信したノード 1 1 は、ツリー 5 2 のルートノードであるノード 1 1 に対して利用タググループ変更通知を送信し、ツリー 5 2 を予備系に遷移させるよう命令する。そして、タググループ 4 1 のフラグを付け、ツリー 5 1 のために送信する B P D U に、現用系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

【 0 3 0 9 】

ツリー 5 1 のノード 1 1 は、利用タググループ変更通知を受信し、ツリー 5 2 を予備系に遷移させ、タググループ 4 2 のフラグを付け、ツリー 5 2 のために送信する B P D U に、予備系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

【 0 3 1 0 】

ノード 1 1 ～ 1 7 は、タググループ 4 1 のタグが付いた B P D U に現用系フラグが付加させたのを確認し、クライアントより転送されてくるフレームに付加するタグを、タググループ 4 2 からタググループ 4 1 に切り替える。このタグが付加されたフレームは、ツリー 5 1 に沿って転送される。

【 0 3 1 1 】

以上の切り替えが完了してしばらく経つと、ツリー 5 2 を流れるフレームはなくなる。

## 【 0 3 1 2 】

ノード 1 7 は、利用タググループ変更通知の送信から一定時間経過後に、タググループ 4 2 のタグを付加するノードが存在しなくなったと判断し、自ノードをネットワークから削除しても良いという、削除許可の通知を、設定インタフェース 1 1 8 に出力する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 3 である。）

## 【 0 3 1 3 】

以降ノードの削除の際は、同様の動作を繰り返す。ただし、上記説明文中のタググループ 4 1 とタググループ 4 2 は適宜入れ替わる形となる。

## 【 0 3 1 4 】

図 1 4 および図 1 5 を参照して、本実施の形態において、タグを B P D U のみに付加し、データにはタグを付加しない場合における、ノード 1 7 を追加する場合の動作について、具体例を用いて詳細に説明する。

## 【 0 3 1 5 】

図 1 4 では、ツリー 5 1 は太線で表され、ノード 1 1 をルートノードとして安定しているとする。安定とは、スパニングツリーのツリー構造が、十分に長い時間変化しない状態になっている状態のことを言う。

## 【 0 3 1 6 】

ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U にはタググループ 4 2 のタグがつく。

## 【 0 3 1 7 】

現在、クライアントからノード 1 1 ～ 1 6 に送信されたデータには、何もタグは付加されない。データは、各ノードのフォワーディングテーブルにおける設定にしたがって、ツリー 5 1 に沿って転送されている。

## 【 0 3 1 8 】

ノード 1 7 は、リンク 2 9 およびリンク 3 0 と接続されると、どのタググループにも加入せず、B P D U の受信を開始する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 1 である。）

## 【 0 3 1 9 】

ノード 1 7 は、各タググループの B P D U を受信すると、B P D U に付加されたタグ内のフラグにより、現時点での現用系がタググループ 4 1 で、現時点での予備系がタググループ 4 2 であると確認する。そして、タググループ 4 2 でのみ B P D U の送受信を行い、タググループ 4 1 では B P D U は受信のみ行い、送信を行わないように設定する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 2 である。）

#### 【 0 3 2 0 】

ノード 1 7 が追加されると、タググループ 4 2 のメンバに変更が発生するため、スパニングツリープロトコルにより、ツリー 5 2 を更新する動作が開始される。タググループ 4 1 のメンバには変更が無いいため、ツリー 5 1 は更新されない。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 3 である。）

#### 【 0 3 2 1 】

クライアントから送信されるフレームには、今までどおりタグは付加されず、ツリー 5 1 に沿って転送され続ける。

#### 【 0 3 2 2 】

ここで、ツリー 5 2 はルートノードをノード 1 1 として、安定したとする。ツリー 5 2 の構成を図 1 5 に示す。

#### 【 0 3 2 3 】

ノード 1 7 は、ネットワークに接続されてから一定時間経過後に、ツリー 5 2 が安定したと判断し、利用タググループ変更通知を、ツリー 5 2 のルートノードである、ノード 1 1 に送信し、ツリー 5 2 を予備系から現用系に遷移させるよう命令する。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 4 である。）

#### 【 0 3 2 4 】

利用タググループ変更通知を受信したツリー 5 2 のノード 1 1 は、ツリー 5 1 のルートノードであるノード 1 1 に対して利用タググループ変更通知を送信し、ツリー 5 1 を予備系に遷移させるよう命令する。そして、タググループ 4 2 のフラグを付け、ツリー 5 2 のために送信する B P D U に、現用系フラグを立てる。



B P D Uは、各ノードで転送されながら、全ノードに伝播する。

【 0 3 2 5 】

ツリー 5 1 のノード 1 1 は、利用タググループ変更通知を受信し、ツリー 5 1 を予備系に遷移させ、タググループ 4 1 のフラグを付け、ツリー 5 1 のために送信する B P D Uに、予備系フラグを立てる。B P D Uは、各ノードで転送されながら、全ノードに伝播する。

【 0 3 2 6 】

ノード 1 1 ～ 1 7 は、タググループ 4 2 のタグが付いた B P D Uに現用系フラグが付加させたのを確認し、ルーティングテーブルを、ツリー 5 2 に従った設定に変更する。これにより、前記ノードにおいて転送されるフレームは、ツリー 5 2 に沿って転送されるようになる。

【 0 3 2 7 】

すべてのノードのルーティングテーブルが、ツリー 5 2 利用のものに切り替わると、ツリー 5 1 を流れるフレームはなくなる。

【 0 3 2 8 】

ノード 1 7 は、利用タググループ変更通知の送信から一定時間経過後に、タググループ 4 1 に従ったテーブル設定を行うノードが存在しなくなったと判断し、自ノードをタググループ 4 1 に加入させ、次のトポロジ変更にも備える。（このときのノード 1 7 のメインコントローラ 1 1 6 4 の状態は、図 1 2 の状態 1 1 6 4 4 である。）

【 0 3 2 9 】

以上のようにして、ネットワークを停止させることなく、ノード 1 7 を追加することができた。以降ノードの追加の際は、同様の動作を繰り返す。ただし、上記説明文中のタググループ 4 1 とタググループ 4 2 は適宜入れ替わる。

【 0 3 3 0 】

以上の説明では、タイマーを用いて、ツリーの安定状態を確認していたが、安定状態の確認方法はこれに限らず、以下に示すように、B P D Uまたはフレームの到着間隔を計測し、それによって安定状態を確認することも可能である。

【 0 3 3 1 】

図 1 4 では、ツリー 5 1 は太線で表され、ノード 1 1 をルートノードとして安定しているとする。

【 0 3 3 2 】

ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U にはタググループ 4 2 のタグがつく。

【 0 3 3 3 】

現在、クライアントからノード 1 1 ～ 1 6 に送信されたデータには、タググループ 4 1 のタグが付加されている。このタグが付加されたデータは、ツリー 5 1 に沿って転送されている。

【 0 3 3 4 】

ノード 1 7 は、リンク 2 9 およびリンク 3 0 と接続されると、どのタググループにも加入せず、B P D U の受信を開始する。

【 0 3 3 5 】

ノード 1 7 は、各タググループの B P D U を受信すると、現時点での現用系がタググループ 4 1 で、現時点での予備系がタググループ 4 2 であると確認する。そして、自ノードをタググループ 4 2 にのみ加入させて B P D U の送受信を行い、タググループ 4 1 では B P D U は受信のみ行い、送信を行わないように設定する。そして、タググループ 4 2 のルートノードであるノード 1 1 および、タググループ 4 1 のルートノードであるノード 1 1 のそれぞれに対して、ノード 1 7 が追加されたことを通知するフレームを送信する。

【 0 3 3 6 】

ノード 1 7 が追加されると、タググループ 4 2 のメンバに変更が発生するため、スパンニングツリープロトコルにより、ツリー 5 2 を更新する動作が開始される。タググループ 4 1 のメンバには変更が無いため、ツリー 5 1 は更新されない。

【 0 3 3 7 】

クライアントから送信されるフレームには、今までどおりタググループ 4 1 のタグが付加され、ツリー 5 1 に沿って転送され続ける。

【 0 3 3 8 】

ここで、ツリー 5 2 はルートノードをノード 1 1 として、安定したとする。ツ

リー 5 2 の構成を図 1 5 に示す。

#### 【 0 3 3 9 】

ツリー 5 2 のルートノードであるノード 1 1 は、ツリー 5 2 の B P D U 到着間隔が一定時間以上となることを検出すると、ツリー 5 2 が安定したと判断し、ツリー 5 1 のルートノードであるノード 1 1 に対して利用タググループ変更通知を送信し、ツリー 5 1 を予備系に遷移させるよう命令する。そして、タググループ 4 2 のフラグを付け、ツリー 5 2 のために送信する B P D U に、現用系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

#### 【 0 3 4 0 】

なお、ツリー 5 2 の安定検出は、ツリー 5 1 のルートノードであるノード 1 1 が行っても良い。この場合、ツリー 5 1 のルートノードであるノード 1 1 は、ツリー 5 2 の B P D U 到着間隔が一定時間以上となることを検出すると、ツリー 5 2 が安定したと判断し、ツリー 5 2 のルートノードであるノード 1 1 に対して利用タググループ変更通知を送信し、ツリー 5 2 を予備系に遷移させるよう命令する。ツリー 5 1 のルートノードであるノード 1 1 は、タググループ 4 2 のフラグを付け、ツリー 5 2 のために送信する B P D U に、現用系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

#### 【 0 3 4 1 】

ノード 1 1 は、利用タググループ変更通知を受信し、ツリー 5 1 を予備系に遷移させ、タググループ 4 1 のフラグを付け、ツリー 5 1 のために送信する B P D U に、予備系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

#### 【 0 3 4 2 】

ノード 1 1 ～ 1 7 は、タググループ 4 2 のタグが付いた B P D U に現用系フラグが付加させたのを確認し、クライアントより転送されてくるフレームに付加するタグを、タググループ 4 1 からタググループ 4 2 に切り替える。このタグが付加されたフレームは、ツリー 5 2 に沿って転送される。

#### 【 0 3 4 3 】

以上の切り替えが完了してしばらく経つと、ツリー 5 1 を流れるフレームはな

くなる。

【 0 3 4 4 】

タググループ 4 1 のルートノードであるノード 1 1 は、タググループ 4 1 のタグを付加してツリー 5 1 を流れるフレームの到着間隔が一定時間以上になると、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、ノード 1 7 に対して G V R P フレームを送信し、タググループ 4 1 に加入するよう命令し、次のトポロジ変更に備える。

【 0 3 4 5 】

なお、ノード 1 7 をタググループ 4 1 に加入させる動作は、ツリー 5 2 のルートノードであるノード 1 1 が行っても良い。

【 0 3 4 6 】

ツリー 5 2 のルートノードであるノード 1 1 が、ノード 1 7 をタググループ 4 1 に加入させる場合は、タググループ 4 1 のタグを付加してツリー 5 1 を流れるフレームの到着間隔が一定時間以上になると、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、ノード 1 7 に対して G V R P フレームを送信し、タググループ 4 1 に加入するよう命令し、次のトポロジ変更に備える。

【 0 3 4 7 】

以上のようにして、ネットワークを停止させることなく、ノード 1 7 を追加することができた。以降ノードの追加の際は、同様の動作を繰り返す。ただし、上記説明文中のタググループ 4 1 とタググループ 4 2 は適宜入れ替わる。

【 0 3 4 8 】

図 1 4 および図 1 5 を参照して、本実施の形態において、予備系への移行完了通知を受信することによって、予備系に遷移したことを検知する場合における、ノード 1 7 を追加する場合の動作について、具体例を用いて詳細に説明する。

【 0 3 4 9 】

図 1 4 では、ツリー 5 1 は太線で表され、ノード 1 1 をルートノードとして安定しているとする。

【 0 3 5 0 】

ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U

にはタググループ 4 2 のタグがつく。

【 0 3 5 1 】

現在、クライアントからノード 1 1 ～ 1 6 に送信されたデータには、タググループ 4 1 のタグが付加されている。このタグが付加されたデータは、ツリー 5 1 に沿って転送されている。

【 0 3 5 2 】

ノード 1 7 は、リンク 2 9 およびリンク 3 0 と接続されると、どのタググループにも加入せず、B P D U の受信を開始する。

【 0 3 5 3 】

ノード 1 7 は、各タググループの B P D U を受信すると、現時点での現用系がタググループ 4 1 で、現時点での予備系がタググループ 4 2 であると確認する。そして、自ノードをタググループ 4 2 にのみ B P D U の送受信を行い、タググループ 4 1 では B P D U は受信のみ行い、送信を行わないように設定する。

【 0 3 5 4 】

ノード 1 7 が追加されると、タググループ 4 2 のメンバに変更が発生するため、スパニングツリープロトコルにより、ツリー 5 2 を更新する動作が開始される。タググループ 4 1 のメンバには変更が無いため、ツリー 5 1 は更新されない。

【 0 3 5 5 】

クライアントから送信されるフレームには、今までどおりタググループ 4 1 のタグが付加され、ツリー 5 1 に沿って転送され続ける。

【 0 3 5 6 】

ここで、ツリー 5 2 はルートノードをノード 1 1 として、安定したとする。ツリー 5 2 の構成を図 1 5 に示す。

【 0 3 5 7 】

ノード 1 7 は、ネットワークに接続されてから一定時間経過後に、ツリー 5 2 が安定したと判断し、利用タググループ変更通知を、ツリー 5 2 のルートノードである、ノード 1 1 に送信し、ツリー 5 2 を予備系から現用系に遷移させるよう命令する。

【 0 3 5 8 】

利用タググループ変更通知を受信したノード 1 1 は、ツリー 5 1 のルートノードであるノード 1 1 に対して利用タググループ変更通知を送信し、ツリー 5 1 を予備系に遷移させるよう命令する。そして、タググループ 4 2 のフラグを付け、ツリー 5 2 のために送信する B P D U に、現用系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

## 【 0 3 5 9 】

ノード 1 1 は、利用タググループ変更通知を受信し、ツリー 5 1 を予備系に遷移させ、タググループ 4 1 のフラグを付け、ツリー 5 1 のために送信する B P D U に、予備系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

## 【 0 3 6 0 】

ノード 1 1 ～ 1 7 は、タググループ 4 2 のタグが付いた B P D U に現用系フラグが付加させたのを確認し、クライアントより転送されてくるフレームに付加するタグを、タググループ 4 1 からタググループ 4 2 に切り替え、さらに切替完了通知をツリー 5 2 のルートノードであるノード 1 1 に対して送信する。

## 【 0 3 6 1 】

以上の切り替えが完了してしばらく経つと、ツリー 5 1 を流れるフレームはなくなる。

## 【 0 3 6 2 】

ノード 1 1 は、ノード 1 1 ～ 1 7 のすべてのノードから切替完了通知を受信すると、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、ノード 1 7 に対して G V R P フレームを送信し、タググループ 4 1 に加入するよう命令する。

## 【 0 3 6 3 】

なお、ノード 1 7 をタググループ 4 1 に加入させる動作は、新規に追加されたノードであるノード 1 7、もしくは、ツリー 5 1 のルートノードであるノード 1 1 が行っても良い。

## 【 0 3 6 4 】

新規に追加されたノードであるノード 1 7 が、ノード 1 7 自身をタググループ



4 1 に加入させる場合は、ノード 1 1 ～ 1 6 が、タググループ 4 2 のタグが付いた B P D U に現用系フラグが付加させたのを確認し、クライアントより転送されてくるフレームに付加するタグを、タググループ 4 1 からタググループ 4 2 に切り替える際に、切替完了通知を新規に追加されたノードであるノード 1 7 に対して送信し、ノード 1 7 が、ノード 1 1 ～ 1 6 のすべてのノードから切替完了通知を受信すると、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、ノード 1 7 自身をタググループ 4 1 に加入させる。

#### 【 0 3 6 5 】

ツリー 5 1 のルートノードであるノード 1 1 が、ノード 1 7 をタググループ 4 1 に加入させる場合は、ノード 1 1 ～ 1 7 が、タググループ 4 2 のタグが付いた B P D U に現用系フラグが付加させたのを確認し、クライアントより転送されてくるフレームに付加するタグを、タググループ 4 1 からタググループ 4 2 に切り替える際に、切替完了通知をツリー 5 1 のルートノードであるノード 1 1 に対して送信し、ノード 1 1 が、ノード 1 1 ～ 1 7 のすべてのノードから切替完了通知を受信すると、タググループ 4 1 のタグを付加するノードが存在しなくなったと判断し、ノード 1 7 に対して G V R P フレームを送信し、タググループ 4 1 に加入するよう命令する。

#### 【 0 3 6 6 】

以上のようにして、ネットワークを停止させることなく、ノード 1 7 を追加することができる。以降ノードの追加の際は、同様の動作を繰り返す。ただし、上記説明文中のタググループ 4 1 とタググループ 4 2 は適宜入れ替わる。

#### 【 0 3 6 7 】

次に、本実施の形態の効果について説明する。

#### 【 0 3 6 8 】

従来、スパニングツリーに属するノードの追加および削除時は、全体もしくは一部のデータフレームの転送を停止させて、スパニングツリーを構築しなおすため、再構築中にネットワークが停止する場合があった。

#### 【 0 3 6 9 】

本実施の形態では、構成変更前のスパニングツリーを運用したまま、新規追加

ノードを含めたスパニングツリーを生成し、新規スパニングツリーの安定後に利用するスパニングツリーを切り替えることにより、ネットワークを停止させずに、スパニングツリーに属するノードの追加および削除等のスパニングツリー再構成が可能である。

【 0 3 7 0 】

またこの結果、輻輳発生の可能性を下げる事が可能である。

【 0 3 7 1 】

(第 2 の実施の形態)

以下、本発明の第 2 の実施の形態について、図面を参照して詳細に説明する。

【 0 3 7 2 】

本発明の第 2 の実施の形態は、第 1 の実施の形態において、コストの計算時にリンク帯域の太さの代わりに、空き帯域の容量もしくは通過TCPフロー数、HTTPリクエスト数等を利用し、さらに、コストが変更された場合には、ノードの追加・削除の場合と同様に、現用系と予備系の遷移を行う点が異なる。なお、以降は空き帯域の容量をコストとして用いた場合について記述するが、通過TCPフロー数およびHTTPリクエスト数についても、特に断りのない限り同様に実現可能である。

【 0 3 7 3 】

IEEE 802.1DおよびIEEE 802.1wでは、リンクのコストはリンク帯域の太さの逆数で決定されていた。つまり、負荷に応じて動的にコストを変化させることができなかった。

【 0 3 7 4 】

本実施の形態においては、リンクのコストをリンクの空き帯域の逆数で決定することで、負荷に応じた動的なコスト変更を行う。

【 0 3 7 5 】

図 1 7 は、第 2 の実施の形態におけるツリーセクタ 1 1 6 の構成を示している。図 1 7 を参照すると、本発明の第 2 の実施の形態は、第 1 の実施の形態における図 1 1 のツリーセクタにおいて、コスト参照タイマ 1 1 6 7、関数演算器 1 1 6 8、および、平滑化回路 1 1 6 9 が追加されている点において異なる。

## 【 0 3 7 6 】

メインコントローラ 1 1 6 4  $\alpha$  は、第 1 の実施における動作のほか、コスト参照タイマ 1 1 6 7 より満了通知を受信すると、リソースモニタから、前回のコスト参照タイマ満了以降にリンクを流れたフレームの流量情報もしくは T C P フロー数または H T T P リクエスト数を取得し、その流量、フロー数、もしくはリクエスト数を元にコストを計算し、予備用に登録されているツリーマネージャ（以下、予備系のツリーマネージャという）に通知する動作を行う。流量の場合は、流量およびリンク帯域の太さより、リンクの空き帯域を求め、前記リングの空き帯域の逆数をコストとして利用する。T C P フロー数および H T T P リクエスト数の場合は、あらかじめ設定してある最大許容フロー数および最大許容リクエスト数と、実際にリンクを通過する T C P フロー数もしくはリクエスト数との差を取り、この差の逆数をコストとして利用する。

## 【 0 3 7 7 】

メインコントローラ 1 1 6 4  $\alpha$  は、前記手段によりコストを算出した後、関数演算器 1 1 6 8 にコストを渡して評価し、さらに関数演算器の評価結果を平滑化回路 1 1 6 9 に渡して平滑化し、平滑化された結果の値を、予備系のツリーマネージャに対して送信する。

## 【 0 3 7 8 】

関数演算器 1 1 6 8 は、メインコントローラ 1 1 6 4  $\alpha$  より入力されたコスト値をパラメータとして、比例関数、ヒステリシス関数、および階段関数等、あらかじめ指定されている任意の関数を用いて、出力コスト値を決定し、メインコントローラ 1 1 6 4 に返す、関数演算器 1 1 6 8 の働きにより、状態遷移の振動を防止することが可能となる。コスト値の急激な変動を抑えコストを滑らかに変化させるためである。

## 【 0 3 7 9 】

平滑化回路 1 1 6 9 は、ローパスフィルタ等を用いて、あらかじめ保存してある前回の入力パラメータと、メインコントローラ 1 1 6 4  $\alpha$  より新たに渡された入力パラメータの平滑化を行い、結果をメインコントローラ 1 1 6 4  $\alpha$  に通知する。平滑化回路 1 1 6 9 の働きにより、コストの急激な変動や、状態遷移の振動

を防止することが可能となる。

【 0 3 8 0 】

コスト参照タイマ 1 1 6 7 は、メインコントローラ 1 1 6 4 α より送信されたセット通知の受信から、あらかじめ設定された時間が経過すると、メインコントローラ 1 1 6 4 α に対してタイマ満了通知を送信する。

【 0 3 8 1 】

図 1 8 は、本発明の第 2 の実施の形態の図 1 7 における、メインコントローラ 1 1 6 4 A の状態遷移を詳細に示した流れ図である。図 1 8 を参照すると、本発明の第 2 の実施の形態は、第 1 の実施の形態における図 1 2 において、状態 1 1 6 4 A および状態 1 1 6 4 B が追加されている点において異なる。

【 0 3 8 2 】

状態 1 1 6 4 A は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 2 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 1 であり、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能が有効にされ、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能も有効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくはは無効に関わらず、常に有効である。

【 0 3 8 3 】

状態 1 1 6 4 B は、現用系ツリーマネージャが、ツリーマネージャ 1 1 5 1 であり、予備系ツリーマネージャがツリーマネージャ 1 1 5 2 であり、ツリーマネージャ 1 1 5 1 内の B P D U 送受信機 1 1 5 1 2 の B P D U 送信機能が有効にされ、ツリーマネージャ 1 1 5 2 内の B P D U 送受信機 1 1 5 2 2 の B P D U 送信機能も有効にされている状態である。なお、B P D U 送受信機 1 1 5 1 2 および B P D U 送受信機 1 1 5 2 2 の B P D U 受信機能は、送信機能の有効もしくはは無効に関わらず、常に有効である。

【 0 3 8 4 】

以下、図 1 8 を参照にして、状態 1 1 6 4 5 を基点としてコスト計算の流れを示すが、この動作は状態 1 1 6 4 6 を基点とした場合においても、同様に適用可

能である。

#### 【 0 3 8 5 】

メインコントローラ 1 1 6 4 α は、状態 1 1 6 4 5 に遷移した際、設定インタフェース 1 1 8 もしくは G V R P 送受信機 1 1 6 2 より、動的なコスト計算を利用する指定を受けた場合に、コスト参照タイマ 1 1 6 7 をセットする。（状態 1 1 6 4 5）

メインコントローラ 1 1 6 4 α は、コスト参照タイマ 1 1 6 7 からタイマ満了の通知を受信すると、リソースモニタ 1 1 9 より累積通過バイト数の情報を受信すると同時に、カウンタリセット通知を出して、リソースモニタ 1 1 9 の累積通過バイト数を 0 にリセットさせる。さらに、前記累積通過バイト数もしくは T C P フロー数または H T T P リクエスト数よりコストを計算し、結果を関数演算器 1 1 6 8 に渡す。

#### 【 0 3 8 6 】

関数演算器 1 1 6 8 は、メインコントローラ 1 1 6 4 α より入力された値を、設定された関数により評価し、結果をメインコントローラ 1 1 6 4 α に返却する。ここでは、比例関数が設定されており、ある入力値に対する出力値が同一になる例をとって解説する。

#### 【 0 3 8 7 】

メインコントローラ 1 1 6 4 α は、関数演算器 1 1 6 8 よりコストの評価結果を受け取ると、その値を平滑化回路 1 1 6 9 に通知する。

#### 【 0 3 8 8 】

平滑化回路 1 1 6 9 は、設定にしたがって入力値をローパスフィルタ等によって平滑化し、結果をメインコントローラ 1 1 6 4 α に返却する。

#### 【 0 3 8 9 】

メインコントローラ 1 1 6 4 α は、平滑化回路 1 1 6 9 より平滑化完了後のコスト値を受信すると、このコスト値を予備系のツリーマネージャ 1 1 5 1 に通知する。ツリーマネージャ 1 1 5 1 は、このコスト情報を元に、スパニングツリーを再計算し、計算の結果、トポロジに変更があるか、ないかを、メインコントローラ 1 1 6 4 α に通知する。（図 1 8 の状態 1 1 6 4 A）

## 【 0 3 9 0 】

メインコントローラ 1 1 6 4 α は、状態 1 1 6 4 A において、再計算後のツリーが計算前のツリーと同一もしくは、あらかじめ設定された変化よりも変化の度合いが小さい場合は、状態 1 1 6 4 5 に遷移し、コスト参照タイマを再セットする。図 1 8 の状態遷移図では、一例として、わずかでも変化がある場合は、状態 1 1 6 4 5 には遷移しない設定であるとして説明している。（状態 1 1 6 4 A）

## 【 0 3 9 1 】

メインコントローラ 1 1 6 4 α は、状態 1 1 6 4 A において、再計算後のツリーが計算前のツリーと異なる場合で、かつ、その変化があらかじめ設定された変化より大きいものであるとき、安定タイマ 1 1 6 5 をセットし、タイマ満了後に状態 1 1 6 4 B に遷移する。図 1 8 の状態遷移図では、わずかでも変化がある場合には、1 1 6 4 B に遷移する設定であるとして説明している。（状態 1 1 6 4 A）

## 【 0 3 9 2 】

メインコントローラ 1 1 6 4 α は、安定タイマ 1 1 6 5 の満了通知を受信すると、現用に登録されているツリーマネージャ 1 1 5 2 と、予備用に登録されているツリーマネージャ 1 1 5 1 を入れ替え、新しく現用をツリーマネージャ 1 1 5 1、予備用をツリーマネージャ 1 1 5 2 とする。さらに、GVRP 送受信機 1 1 6 2 を通じて、新ツリーのルートノードに対して、利用タググループ変更通知を送信する。この利用タググループ変更通知の内容は、BPDU に反映され、全ノードに伝達される。その後、安定タイマ 1 1 6 5 を作動させ、状態 1 1 6 4 6 に遷移する。（状態 1 1 6 4 B）

## 【 0 3 9 3 】

メインコントローラ 1 1 6 4 α は、安定タイマ 1 1 6 5 の満了通知を受信すると、ツリーマネージャ 1 1 5 2 に対して新規コストの通知を行い、新たに計算したコスト情報を元に、スパニングツリーを再計算させる。また、コスト参照タイマ 1 1 6 7 をスケジュールする。（状態 1 1 6 4 6）

## 【 0 3 9 4 】

次に図 7 を利用して、本実施の形態において、ノード 1 5 からノード 1 3 への



転送経路が変更される場合の、スパニングツリー切り替え動作について、具体例を用いて詳細に説明する。

【 0 3 9 5 】

図 7 を参照すると、本動作例では、ノード 1 1 ～ 1 6、クライアント 9 1 ～ 9 6、双方向リンク 8 1 ～ 8 6、および、双方向リンク 2 1 ～ 2 8 を含む。

【 0 3 9 6 】

ノード 1 1 にはリンク 8 1 によりクライアント 9 1 が、ノード 1 2 にはリンク 8 2 によりクライアント 9 2 が、ノード 1 3 にはリンク 8 3 によりクライアント 9 3 が、ノード 1 4 にはリンク 8 4 によりクライアント 9 4 が、ノード 1 5 にはリンク 8 5 によりクライアント 9 5 が、ノード 1 6 にはリンク 2 6 によりクライアント 9 6 が、それぞれ接続されている。

【 0 3 9 7 】

ノード 1 1 とノード 1 2 の間はリンク 2 1 で、ノード 1 2 とノード 1 3 の間はリンク 2 2 で、ノード 1 3 とノード 1 4 の間はリンク 2 3 で、ノード 1 1 とノード 1 5 の間はリンク 2 4 で、ノード 1 5 とノード 1 6 の間はリンク 2 5 で、ノード 1 6 とノード 1 4 の間はリンク 2 6 で、ノード 1 2 とノード 1 5 の間はリンク 2 7 で、ノード 1 3 とノード 1 6 の間はリンク 2 8 で、それぞれ接続されている。

【 0 3 9 8 】

ノード 1 1 ～ 1 6 の全ポートが所属するタググループが 2 つすでに設定されており、1 つめのタググループをタググループ 4 1、2 つめのタググループをタググループ 4 2 と呼ぶ。

【 0 3 9 9 】

ノード 1 1 ～ 1 6 には、独立して動作するスパニングツリー回路が 2 つ存在し、タググループ 4 1 上で動作するスパニングツリーをツリー 5 1、タググループ 4 2 上で動作するスパニングツリーをツリー 5 2 と呼ぶ。

【 0 4 0 0 】

図 7 において、ツリー 4 1 は、ノード 1 3 をルートノードとし、すべてのリンクに等しく 1 0 に設定された初期コストを用いて、すでに安定しているとする。

ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U にはタググループ 4 2 のタグがつく。

【 0 4 0 1 】

現在、クライアントからノード 1 1 ~ 1 6 に送信されたデータには、現用系であるタググループ 4 1 のタグが付加されている。このタグが付加されたデータは、ツリー 5 1 に沿って転送されている。

【 0 4 0 2 】

初期状態では、クライアント 9 1 ~ 9 6 のうち、どのクライアントもデータ転送を行っていない。

【 0 4 0 3 】

各ノードは、一定時間ごとに、H e l l o T i m e で規定される周期で B P D U を送受信することにより、B P D U の状態を確認している。このフレームには、現用系と予備系の識別フラグがあり、現在は現用系であるタググループ 4 1 のタグが付加された B P D U のみに、現用系のフラグがつき、予備系であるタググループ 4 2 のタグが付加された B P D U には、現用系を示すフラグは付加されない。

【 0 4 0 4 】

各ノードは、設定インタフェース 1 1 8 もしくは G V R P 送受信機 1 1 6 2 より、すでに動的なコスト計算を利用する指定を受けているため、コスト参照タイマが満了するごとに、前回のタイマ満了以降にリンクを流れたフレームの流量を参照し、コストの再計算および、予備系ツリーを用いたスパニングツリーの再計算を行う。

【 0 4 0 5 】

ここで、クライアント 9 5 からクライアント 9 3、および、クライアント 9 6 からクライアント 9 3 との間で、データ転送を開始したとする。

【 0 4 0 6 】

転送初期においては、クライアント 9 5 からクライアント 9 3 へのデータは、ツリー 5 1 を利用し、リンク 8 5、2 5、2 8、8 3 を経由して転送される。また、クライアント 9 6 からクライアント 9 3 へのデータも、ツリー 5 1 を利用し

、リンク 8 6、2 8、8 3 を経由して転送される。

【 0 4 0 7 】

データ転送から一定時間が経過すると（各ノードのコスト参照タイマ 1 1 6 7 が満了すると）、リンク 2 1 ～ 2 8 の空き容量をもとに、ツリー 5 2 におけるリンク 2 1 ～ 2 8 のコストが再計算され、関数演算器および平滑化回路にかけられる。ここで、リンク 2 8 の空き帯域は少なくなっているため、ノード 1 6 によりツリー 5 2 のリンク 2 8 のコストは 1 5 に、リンク 2 5 の空き帯域は、リンク 2 8 ほどではないが少なくなっているため、ノード 1 5 によりツリー 5 2 のリンク 2 5 のコストは 1 2 に、それぞれ変更されたとする。このとき、現在使用している、ツリー 5 1 のコストは変更されない。

【 0 4 0 8 】

ノード 1 6 は、コスト変更を検知し、変更後のコストを利用して作成した B P D U（図 5 の Root path Cost 22056 の値が変更される）を、隣接するノード 1 3、ノード 1 4、ノード 1 5 に対して、それぞれ送信する。この B P D U にはタググループ 4 2 のタグが付加される。

【 0 4 0 9 】

ノード 1 5 も同様に、コスト変更を検知し、変更後のコストを利用して作成した B P D U を、隣接するノード 1 1、ノード 1 2、ノード 1 6 に対して、それぞれ送信する。

【 0 4 1 0 】

ノード 1 5 は、ノード 1 6 からコスト 1 5 が付加された B P D U を受信すると、リンク 2 5 のコストも加算し、リンク 2 5、リンク 2 8 経由でノード 1 3 に到達するために、コスト 2 7 がかかることを認識する。

【 0 4 1 1 】

その後ノード 1 5 は、ノード 1 2 から、コスト 1 0 が付加された B P D U を受信すると（ノード 1 2 は、ルートノード（ノード 1 3）までのコストを、定期的に隣接ノード（ノード 1 1、およびノード 1 5）に送信する）、リンク 2 7 のコストも加算し、リンク 2 7、リンク 2 2 経由でノード 1 3 に到達するために、コスト 2 0 がかかることを認識する。このコストは、リンク 2 5 経由のコストより

も小さいため、ノード 1 5 は停止ポートをリンク 2 7 側からリンク 2 5 側に切り替え、リンク 2 7 およびリンク 2 2 を利用して、リンク 1 3 に到達するツリーを形成する。リンク 2 7、2 2 経由のツリーを作成するときにもスパニングツリープロトコルが使用される。そして、安定タイマ 1 1 6 5 を起動し、ツリーが安定するのを待つ。

## 【 0 4 1 2 】

ここで、ツリー 5 2 が、ノード 1 3 をルートノードとして安定したとする。

## 【 0 4 1 3 】

ツリー 5 1 のルートノードであるノード 1 3 は、図 1 3 の安定タイマ 1 1 6 5 が満了すると、ツリー形成が安定したと判断し、新ツリーとなるツリー 5 2 のルートノードであるノード 1 3 に対して、転送に利用する現用系ツリーをツリー 5 1 からツリー 5 2 に切り替えるよう命令する、タググループ変更メッセージを送信する。その後、安定タイマ 1 1 6 5 を作動させる。

## 【 0 4 1 4 】

ツリー 5 2 のルートノードであるノード 1 3 は、ツリー切替メッセージを受信すると、旧現用系ツリーであるツリー 4 1 のルートであるノード 1 3 に対して、転送に利用する現用系ツリーをツリー 5 1 からツリー 5 2 に切り替えるよう命令する、タググループ変更メッセージを送信する。この利用タググループ変更通知の内容は、ノード 1 3 から送信される、ツリー 5 1 およびツリー 5 2 の各 B P D U に反映され、全ノードに伝達される。

## 【 0 4 1 5 】

ノード 1 3 は、旧現用系ルートノードへのタググループ変更メッセージの送信を完了させると、自ノードがクライアント 9 3 より受信して、ネットワーク内に送信するフレームの転送に利用するツリーを、今まで利用していたツリー 5 1 からツリー 5 2 へ切り替える。切り替えが完了すると、クライアント 9 3 からクライアント 9 5 に向けて送信されていたフレームは、リンク 8 3、リンク 2 2、リンク 2 7、リンク 8 5 を経由して、クライアント 9 5 に転送されるようになる。

## 【 0 4 1 6 】

このようにして、クライアント 9 3 からクライアント 9 5、および、クライア

ント 9 3 からクライアント 9 6 に向けて送信されるフレームの転送経路は分散され、リンク 2 8 の混雑が解消される。

【 0 4 1 7 】

以降、コスト参照タイマが満了するごとに、リンクの空き帯域に基づくコスト計算によってスパニングツリーが再計算され、周期的に空き帯域をコストに反映する動的な経路変更が行われる。この結果、各リンクのトラフィックが分散され、リンクの負荷を分散し、輻輳を防止することができる。

【 0 4 1 8 】

次に、本実施の形態の効果について説明する。

【 0 4 1 9 】

従来、リンク容量を用いてコストを計算し、スパニングツリー構築時の経路選択に利用しているため、トラフィックに応じた動的な負荷分散のための経路変更ができなかった。

【 0 4 2 0 】

本実施の形態では、リンクコストを、空き帯域やサーバ負荷等の動的情報によって計算することにより、トラフィックの負荷分散が可能である。

【 0 4 2 1 】

また従来、コストをトラフィック状況に応じて動的に変化させようとする、局所的もしくはネットワーク全体で、データフレームの転送を停止させてスパニングツリーを構築しなおして経路を変更するため、再構築中にネットワークが停止する場合があった。

【 0 4 2 2 】

本実施の形態では、変更前のツリーを運用したまま、コスト変更後のツリーを生成し、新規ツリーの安定後に利用するツリーを切り替えることにより、経路変更にもなうスパニングツリー再構成のために、ネットワークを停止させることなく負荷分散が可能である。

またこの結果、輻輳発生の可能性を下げることも可能である。

【 0 4 2 3 】

(第 3 の実施の形態)

以下、本発明の第 3 の実施の形態について、図面を参照して詳細に説明する。

【 0 4 2 4 】

本発明の第 3 の実施の形態は、第 2 の実施の形態において、コストが変更されたかされなかったかに関わらず、現用系と予備系の遷移を行う点が異なる。なお、以降は空き帯域の容量をコストとして用いた場合について記述するが、通過 TCP フロー数および HTTP リクエスト数についても、特に断りのない限り同様に実現可能である。

【 0 4 2 5 】

図 1 9 を参照すると、本発明の第 3 の実施の形態は、第 2 の実施の形態における図 1 8 において、状態 1 1 6 4 A と状態 1 1 6 4 B の間の遷移が発生しないこと、ならびに、状態 1 1 6 4 3 と状態 1 1 6 4 4 の間の遷移および、状態 1 1 6 4 7 と状態 1 1 6 4 8 の間の遷移が、安定タイマ 1 1 6 5 の満了によって起こるのではなく、B P D U 中の現用系フラグの切り替え検出によって発生する点が異なる。

【 0 4 2 6 】

第 3 の実施の形態におけるメインコントローラ 1 1 6 4 β は、リソースモニタ 1 1 9 より新たにネットワークに接続されたことの通知を受信すると、ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 からの現用系通知の到着を待つ。ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 より B P D U に含まれる現用系通知を受信すると、前記通知において指定されたツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 を現用に、予備用に指定されたツリーマネージャ 1 1 5 1 もしくは 1 1 5 2 を予備用に設定し、状態 1 1 6 4 2 もしくは状態 1 1 6 4 9 に遷移する。ここでは、状態 1 1 6 4 2 に遷移した場合を例に解説するが、以後の解説は状態 1 1 6 4 9 に遷移した場合においても同様である。（図 1 9 の状態 1 1 6 4 1）

【 0 4 2 7 】

メインコントローラ 1 1 6 4 β は、ツリーマネージャ 1 1 5 1 を現用、ツリーマネージャ 1 1 5 2 を予備用に設定する。さらに、前記各ツリーマネージャ 1 1 5 1 および 1 1 5 2 に対して、B P D U 送信停止命令を出す。（図 1 9 の状態 1



1 6 4 2)

【 0 4 2 8 】

メインコントローラ 1 1 6 4  $\beta$  は、状態 1 1 6 4 2 において、もし設定インタフェース 1 1 8 よりノードの追加要求を受信した場合は、状態 1 1 6 4 3 に遷移する。また、もしツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 より現用系通知を受信し、現用系と予備系の関係に変更があった場合には、状態 1 1 6 4 9 に遷移する。(図 1 9 の状態 1 1 6 4 2)

【 0 4 2 9 】

メインコントローラ 1 1 6 4  $\beta$  は、ツリーマネージャ 1 1 5 2 に対してリンク up の通知を行い、同時に、ツリーマネージャ 1 1 5 2 に対して B P D U の送信許可を行う。さらに、安定タイマ 1 1 6 5 を作動させる。(図 1 9 の状態 1 1 6 4 3)

メインコントローラ 1 1 6 4  $\beta$  は、状態 1 1 6 4 3 において、ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 より B P D U に含まれる現用系通知を受信し、もし現用系と予備系の関係に変更が生じた場合には、現用に登録されているツリーマネージャ 1 1 5 1 と、予備用に登録されているツリーマネージャ 1 1 5 2 を入れ替え、新しく現用をツリーマネージャ 1 1 5 2、予備用をツリーマネージャ 1 1 5 1 とする。その後、安定タイマ 1 1 6 5 を作動させる。(図 1 9 の状態 1 1 6 4 4)

【 0 4 3 0 】

メインコントローラ 1 1 6 4  $\beta$  は、安定タイマ 1 1 6 5 よりタイマ満了通知を受信すると、ツリーマネージャ 1 1 5 1 に対してリンク up の通知を行い、同時に、ツリーマネージャ 1 1 5 2 に対して B P D U の送信許可を行う。通常は、この状態で安定する。(図 1 9 の状態 1 1 6 4 5)

【 0 4 3 1 】

状態 1 1 6 4 5 において、もし現用であるツリーマネージャ 1 1 5 2 がルートノードとなっている場合、このノードのメインコントローラ 1 1 6 4  $\beta$  は、安定タイマ 1 1 6 5 を動作させ、安定タイマ 1 1 6 5 よりタイマ満了通知を受信すると、現用に登録されているツリーマネージャ 1 1 5 2 と、予備用に登録されてい

るツリーマネージャ1151を入れ替え、新しく現用をツリーマネージャ1151、予備用をツリーマネージャ1152とする。さらに、GVRP送受信機1162を通じて、新ツリーのルートノードに対して、利用タググループ変更通知を送信する。この利用タググループ変更通知の内容は、BPDUに反映され、全ノードに伝達される。(図19の状態11645)

#### 【0432】

前記ツリーマネージャの入れ替えは、予備用であるツリーマネージャ1151が行っても良い。この場合、状態11645において、もし予備用であるツリーマネージャ1151がルートノードとなっている場合、ノードのメインコントローラ1164は、安定タイマ1165を動作させ、安定タイマ1165よりタイマ満了通知を受信すると、現用に登録されているツリーマネージャ1152と、予備用に登録されているツリーマネージャ1151を入れ替え、新しく現用をツリーマネージャ1151、予備用をツリーマネージャ1152とする。さらに、GVRP送受信機1162を通じて、旧現用ツリーのルートノードに対して、利用タググループ変更通知を送信する。この利用タググループ変更通知の内容は、BPDUに反映され、全ノードに伝達される。(図19の状態11645)

#### 【0433】

メインコントローラ1164 $\beta$ は、状態11645において、ツリーマネージャ1151もしくはツリーマネージャ1152よりBPDUに含まれる現用系通知を受信し、もし現用系と予備系の関係に変更が生じた場合には、状態11646に遷移する。そして、現用に登録されているツリーマネージャ1152と、予備用に登録されているツリーマネージャ1151を入れ替え、新しく現用をツリーマネージャ1151、予備用をツリーマネージャ1152とする。(図19の状態11645)

#### 【0434】

メインコントローラ1164 $\beta$ は、状態11645において、もし設定インタフェース118より、ノード削除要求を受信した場合は、状態11644に遷移する。(図19の状態11645)

#### 【0435】

メインコントローラ 1 1 6 4  $\beta$  は、ツリーマネージャ 1 1 5 1 に対して、接続されている全リンクリンク d o w n の通知を行い、同時に、ツリーマネージャ 1 1 5 1 に対して B P D U の送信停止命令を出す。（図 1 9 の状態 1 1 6 4 4）

## 【 0 4 3 6 】

メインコントローラ 1 1 6 4  $\beta$  は、状態 1 1 6 4 4 において、ツリーマネージャ 1 1 5 1 もしくはツリーマネージャ 1 1 5 2 より B P D U に含まれる現用系通知を受信し、もし現用系と予備系の関係に変更が生じた場合には、状態 1 1 6 4 3 に遷移し、現用に登録されているツリーマネージャ 1 1 5 2 と、予備用に登録されているツリーマネージャ 1 1 5 1 を入れ替え、新しく現用をツリーマネージャ 1 1 5 1、予備用をツリーマネージャ 1 1 5 2 とする。その後、安定タイマ 1 1 6 5 を作動させる。（図 1 9 の状態 1 1 6 4 3）

## 【 0 4 3 7 】

メインコントローラ 1 1 6 4  $\beta$  は、安定タイマ 1 1 6 5 よりタイマ満了通知を受信すると、ツリーマネージャ 1 1 5 1 に対してリンク d o w n の通知を行い、同時に、ツリーマネージャ 1 1 5 1 に対して B P D U の受信停止命令を出す。さらに、無条件で状態 1 1 6 4 1 に遷移し、ノード切離しまで待機する。（図 1 9 の状態 1 1 6 4 2）

## 【 0 4 3 8 】

メインコントローラ 1 1 6 4  $\beta$  は、設定インタフェース 1 1 8 もしくは G V R P 送受信機 1 1 6 2 より、動的なコスト計算を利用する指定を受けた場合には、利用タググループ変更通知の受信から一定時間経過後に満了となるよう、コスト参照タイマ 1 1 6 7 をセットする。（図 1 9 の状態 1 1 6 4 5）

## 【 0 4 3 9 】

メインコントローラ 1 1 6 4  $\beta$  は、コスト参照タイマ 1 1 6 7 からタイマ満了の通知を受信すると、リソースモニタ 1 1 9 より累積通過バイト数の情報を受信すると同時に、カウンタリセット通知を出して、リソースモニタ 1 1 9 の累積通過バイト数を 0 にリセットさせる。さらに、前記累積通過バイト数もしくは T C P フロー数または H T T P リクエスト数よりコストを計算し、予備系であるツリーマネージャ 1 1 5 1 通知する。ツリーマネージャ 1 1 5 1 は、新たに計算した

コスト情報を元に、スパニングツリーを再計算し、計算の結果、トポロジに変更があるか、ないかを、メインコントローラ 1 1 6 4  $\beta$  に通知する。そして、無条件で状態 1 1 6 4 5 に遷移する。（図 1 9 の状態 1 1 6 4 A）

【 0 4 4 0 】

図 1 4 および図 1 5 を参照して、本実施の形態において、ノード 1 7 を追加する場合の動作について、具体例を用いて詳細に説明する。

【 0 4 4 1 】

図 1 4 を参照すると、本動作例では、ノード 1 1 ～ 1 7、およびリンク 2 1 ～ 3 0 を有する。ただし、ノード 1 7、リンク 2 9 およびリンク 3 0 は、初期状態では接続されていない。

【 0 4 4 2 】

図 1 5 を参照すると、本動作例では、ノード 1 1 ～ 1 7、およびリンク 2 1 ～ 3 0 を有する。

【 0 4 4 3 】

初期状態でノード 1 1 ～ 1 6 の全ノード、全ポートが所属するタググループが 2 つすでに設定されており、1 つめのタググループをタググループ 4 1、2 つめのタググループをタググループ 4 2 と呼ぶ。

【 0 4 4 4 】

なお、基本的には全ノード、全ポートを 2 つのタググループに加入させるが、一部のポートもしくはノードのみで構成させるタググループを作成しても良い。以降、全ノード、全ポートが 2 つのタググループに加入するものとして説明する。

【 0 4 4 5 】

ノード 1 1 ～ 1 6 には、独立して動作するスパニングツリー回路が 2 つ存在し、タググループ 4 1 上で動作するスパニングツリーをツリー 5 1、タググループ 4 2 上で動作するスパニングツリーをツリー 5 2 と呼ぶ。

【 0 4 4 6 】

スパニングツリーは必ず 2 系統作成するが、タググループは必ずしも 2 つ作成する必要は無い。前記タググループ 4 1 のみ設定してタググループ 4 2 を利用せ

ず、タググループ 4 1 上で動作するスパニングツリーをツリー 5 1、タググループに属さないで動作するスパニングツリーをツリー 5 2 と呼ぶこともできる。また逆に、前記タググループ 4 2 のみ設定してタググループ 4 1 を利用せず、タググループに属さないで動作するスパニングツリーをツリー 5 1、タググループ 4 2 上で動作するスパニングツリーをツリー 5 2 と呼ぶこともできる。

## 【 0 4 4 7 】

ここでは、特にタググループ 4 1 とタググループ 4 2 の両方を用いた場合について説明を行うが、タググループ 4 1 及びタググループ 4 2 の両方を用いた場合の動作は、タググループ 4 1 のみ、もしくはタググループ 4 2 のみを用いた場合においても、同様に適用可能である。

## 【 0 4 4 8 】

図 1 4 では、ツリー 5 1 は太線で表され、ノード 1 1 をルートノードとして安定しているとする。

## 【 0 4 4 9 】

ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U にはタググループ 4 2 のタグがつく。

## 【 0 4 5 0 】

現在、クライアントからノード 1 1 ～ 1 6 に送信されたデータには、タググループ 4 1 のタグが付加されている。このタグが付加されたデータは、ツリー 5 1 に沿って転送されている。

## 【 0 4 5 1 】

ノード 1 7 は、リンク 2 9 およびリンク 3 0 と接続されると、どのタググループにも加入せず、B P D U の受信を開始する。

## 【 0 4 5 2 】

ノード 1 7 は、各タググループの B P D U を受信すると、現時点での現用系がタググループ 4 1 で、現時点での予備系がタググループ 4 2 であると確認する。そして、自ノードをタググループ 4 2 にのみ参加させて B P D U の送受信を行い、タググループ 4 1 では B P D U は受信のみ行い、送信を行わないように設定する。

## 【 0 4 5 3 】

ノード 1 7 が追加されると、タググループ 4 2 のメンバに変更が発生するため、スパニングツリープロトコルにより、ツリー 5 2 を更新する動作が開始される。タググループ 4 1 のメンバには変更が無いため、ツリー 5 1 は更新されない。

## 【 0 4 5 4 】

クライアントから送信されるフレームには、今までどおりタググループ 4 1 のタグが付加され、ツリー 5 1 に沿って転送され続ける。

## 【 0 4 5 5 】

現用系ツリーであるツリー 5 1 のルートノードとなっているノード 1 1 は、安定タイマが満了すると、ツリー 5 1 のルートノードであるノード 1 1 に対して利用タググループ変更通知を送信し、ツリー 5 1 を予備系に遷移させるよう命令する。そして、タググループ 4 2 のフラグを付け、ツリー 5 2 のために送信する B P D U に、現用系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

## 【 0 4 5 6 】

ノード 1 1 は、現用系フラグの付加状況の変更を感知し、ツリー 5 1 を予備系に遷移させ、タググループ 4 1 のフラグを付け、ツリー 5 1 のために送信する B P D U に、予備系フラグを立てる。B P D U は、各ノードで転送されながら、全ノードに伝播する。

## 【 0 4 5 7 】

ノード 1 1 ～ 1 7 は、タググループ 4 2 のタグが付いた B P D U に現用系フラグが付加させたのを確認し、クライアントより転送されてくるフレームに付加するタグを、タググループ 4 1 からタググループ 4 2 に切り替える。このタグが付加されたフレームは、ツリー 5 2 に沿って転送される。

## 【 0 4 5 8 】

以上の切り替えが完了してしばらく経つと、ツリー 5 1 を流れるフレームはなくなる。

## 【 0 4 5 9 】

ノード 1 7 は、利用タググループ変更通知の送信から一定時間経過後に、タグ



グループ 4 1 のタグを付加するノードが存在しなくなったと判断し、自ノードをタググループ 4 1 に加入させ、次のトポロジ変更に備える。

#### 【 0 4 6 0 】

以上のようにして、ネットワークを停止させることなく、ノード 1 7 を追加することができた。以降ノードの追加の際は、同様の動作を繰り返す。ただし、上記説明文中のタググループ 4 1 とタググループ 4 2 は入れ替わる。

#### 【 0 4 6 1 】

次に図 7 を利用して、本実施の形態において、ノード 1 5 からノード 1 3 への転送経路が変更される場合の、スパニングツリー切り替え動作について、具体例を用いて詳細に説明する。

#### 【 0 4 6 2 】

図 7 を参照すると、本動作例では、ノード 1 1 ～ 1 6、クライアント 9 1 ～ 9 6、双方向リンク 8 1 ～ 8 6、および、双方向リンク 2 1 ～ 2 8 を含む。

#### 【 0 4 6 3 】

ノード 1 1 にはリンク 8 1 によりクライアント 9 1 が、ノード 1 2 にはリンク 8 2 によりクライアント 9 2 が、ノード 1 3 にはリンク 8 3 によりクライアント 9 3 が、ノード 1 4 にはリンク 8 4 によりクライアント 9 4 が、ノード 1 5 にはリンク 8 5 によりクライアント 9 5 が、ノード 1 6 にはリンク 2 6 によりクライアント 9 6 が、それぞれ接続されている。

#### 【 0 4 6 4 】

ノード 1 1 とノード 1 2 の間はリンク 2 1 で、ノード 1 2 とノード 1 3 の間はリンク 2 2 で、ノード 1 3 とノード 1 4 の間はリンク 2 3 で、ノード 1 1 とノード 1 5 の間はリンク 2 4 で、ノード 1 5 とノード 1 6 の間はリンク 2 5 で、ノード 1 6 とノード 1 4 の間はリンク 2 6 で、ノード 1 2 とノード 1 5 の間はリンク 2 7 で、ノード 1 3 とノード 1 6 の間はリンク 2 8 で、それぞれ接続されている。

#### 【 0 4 6 5 】

ノード 1 1 ～ 1 6 の全ポートが所属するタググループが 2 つすでに設定されており、1 つめのタググループをタググループ 4 1、2 つめのタググループをタグ

グループ 4 2 と呼ぶ。

【 0 4 6 6 】

ノード 1 1 ～ 1 6 には、独立して動作するスパニングツリー回路が 2 つ存在し、タググループ 4 1 上で動作するスパニングツリーをツリー 5 1、タググループ 4 2 上で動作するスパニングツリーをツリー 5 2 と呼ぶ。

【 0 4 6 7 】

ツリー 4 1 は太線で表され、ノード 1 3 をルートノードとし、すべてのリンクに等しく 1 0 に設定された初期コストを用いて、すでに安定しているとする。ツリー 5 1 の B P D U にはタググループ 4 1 のタグが、ツリー 5 2 の B P D U にはタググループ 4 2 のタグがつく。

【 0 4 6 8 】

現在、クライアントからノード 1 1 ～ 1 6 に送信されたデータには、現用系であるタググループ 4 1 のタグが付加されている。このタグが付加されたデータは、ツリー 5 1 に沿って転送されている。

【 0 4 6 9 】

初期状態では、クライアント 9 1 ～ 9 6 のうち、どのクライアントもデータ転送を行っていない。

【 0 4 7 0 】

各ノードは、一定時間ごとに、H E L L O フレームを送信し、B P D U の状態を確認している。このフレームには、現用系と予備系の識別フラグがあり、現在は現用系であるタググループ 4 1 のタグが付加された B P D U のみに、現用系のフラグがつき、予備系であるタググループ 4 2 のタグが付加された B P D U には、現用系を示すフラグは付加されない。

【 0 4 7 1 】

各ノードは、設定インタフェース 1 1 8 もしくは G V R P 送受信機 1 1 6 2 より、すでに動的なコスト計算を利用する指定を受けているため、コスト参照タイマが満了するごとに、前回のタイマ満了以降にリンクを流れたフレームの流量を参照し、コストの再計算および、予備系ツリーを用いたスパニングツリーの再計算を行う。

## 【 0 4 7 2 】

ここで、クライアント 9 5 からクライアント 9 3、および、クライアント 9 6 からクライアント 9 3 との間で、データ転送を開始したとする。

## 【 0 4 7 3 】

転送初期においては、クライアント 9 5 からクライアント 9 3 へのデータは、ツリー 5 1 を利用し、リンク 8 5、2 5、2 8、8 3 を経由して転送される。また、クライアント 9 6 からクライアント 9 3 へのデータも、ツリー 5 1 を利用し、リンク 8 6、2 8、8 3 を経由して転送される。

## 【 0 4 7 4 】

データ転送から一定時間が経過すると、リンク 2 1 ～ 2 8 の空き容量をもとに、ツリー 5 2 におけるリンク 2 1 ～ 2 8 のコストが片側ずつ再計算される。ここで、リンク 2 8 の空き帯域は少なくなっているため、ノード 1 6 によりツリー 5 2 のリンク 2 8 のコストは 1 5 に、リンク 2 5 の空き帯域は、リンク 2 8 ほどではないが少なくなっているため、ノード 1 5 によりツリー 5 2 のリンク 2 5 のコストは 1 2 に、それぞれ変更されたとする。このとき、現在使用している、ツリー 5 1 のコストは変更されない。

## 【 0 4 7 5 】

ノード 1 6 は、コスト変更を検知し、変更後のコストを利用して作成した B P D U を、隣接するノード 1 3、ノード 1 4、ノード 1 5 に対して、それぞれ送信する。

## 【 0 4 7 6 】

ノード 1 5 も同様に、コスト変更を検知し、変更後のコストを利用して作成した B P D U を、隣接するノード 1 1、ノード 1 2、ノード 1 6 に対して、それぞれ送信する。

## 【 0 4 7 7 】

ノード 1 5 は、ノード 1 6 からコスト 1 5 が付加された B P D U を受信すると、リンク 2 5 のコストも加算し、リンク 2 5、リンク 2 8 経由でノード 1 3 に到達するために、コスト 2 7 がかかることを認識する。

## 【 0 4 7 8 】

その後ノード 1 5 は、ノード 1 2 から、コスト 1 0 が付加された B P D U を受信すると、リンク 2 7 のコストも加算し、リンク 2 7、リンク 2 2 経由でノード 1 3 に到達するために、コスト 2 0 がかかることを認識する。このコストは、リンク 2 5 経由のコストよりも小さいため、ノード 1 5 は停止ポートをリンク 2 7 側からリンク 2 5 側に切り替え、リンク 2 7 およびリンク 2 2 を利用して、リンク 1 3 に到達するツリーを形成する。

## 【 0 4 7 9 】

ツリー 5 1 のルートノードであるノード 1 3 において起動していたトポロジ安定タイマが満了すると、旧現用系ツリーであるツリー 4 1 のルートであるノード 1 3 に対して、転送に利用する現用系ツリーをツリー 5 1 からツリー 5 2 に切り替えるよう命令する、タググループ変更メッセージを送信する。この利用タググループ変更通知の内容は、ノード 1 3 から送信される、ツリー 5 1 およびツリー 5 2 の各 B P D U に反映され、全ノードに伝達される。

## 【 0 4 8 0 】

ノード 1 3 は、旧現用系ルートノードへのタググループ変更メッセージの送信を完了させると、自ノードがクライアント 9 3 より受信して、ネットワーク内に送信するフレームの転送に利用するツリーを、今まで利用していたツリー 5 1 からツリー 5 2 へ切り替える。切り替えが完了すると、クライアント 9 3 からクライアント 9 5 に向けて送信されていたフレームは、リンク 8 3、リンク 2 2、リンク 2 7、リンク 8 5 を経由して、クライアント 9 5 に転送されるようになる。

## 【 0 4 8 1 】

このようにして、クライアント 9 3 からクライアント 9 5、および、クライアント 9 3 からクライアント 9 6 に向けて送信されるフレームの転送経路は分散され、リンク 2 8 の混雑が解消される。

## 【 0 4 8 2 】

以降、コスト参照タイマが満了するごとに、リンクの空き帯域に基づくコスト計算によってスパニングツリーが再計算され、周期的に空き帯域をコストに反映する動的な経路変更が行われる。この結果、各リンクのトラフィックが分散され、リンクの負荷を分散し、輻輳を防止することができる。

【 0 4 8 3 】

次に、本実施の形態の効果について説明する。

【 0 4 8 4 】

従来、リンク容量を用いてコストを計算し、スパニングツリー構築時の経路選択に利用しているため、トラフィックに応じた動的な負荷分散のための経路変更ができなかった。

【 0 4 8 5 】

本実施の形態では、リンクコストを、空き帯域やサーバ負荷等の動的情報によって計算することにより、トラフィックの負荷分散が可能である。

【 0 4 8 6 】

また従来、コストをトラフィック状況に応じて動的に変化させるたびに、局所的もしくはネットワーク全体で、データフレームの転送を停止させてスパニングツリーを構築しなおして経路を変更するため、再構築中にネットワークが停止する場合があった。

【 0 4 8 7 】

本実施の形態では、変更前のツリーを運用したまま、コスト変更後のツリーを生成し、新規ツリーの安定後に利用するツリーを切り替えることにより、経路変更にもなうスパニングツリー再構成のために、ネットワークを停止させることなく負荷分散が可能である。

またこの結果、輻輳発生の可能性を下げることが可能である。

【 0 4 8 8 】

(第 4 の実施の形態)

以下、本発明の第 4 の実施の形態について、図面を参照して詳細に説明する。

【 0 4 8 9 】

本発明の第 4 の実施の形態は、第 1 の実施の形態において、宛先ノード別に利用するタグおよびスパニングツリーを切り替え、さらに、宛先のノードをルートノードに設定する場合に対応する。

【 0 4 9 0 】

IEEE 802.1D および IEEE 802.1w が動作しているネットワー

クにおいてフレームを送信した場合、宛先への最低コスト経路が必ずしも選択されるとは限らず、使用されないリンクが出現するほか、ルートノードに負荷が集中する、ルートノード障害時に長時間ネットワークが停止するなどの問題があった。

## 【 0 4 9 1 】

本実施の形態においては、宛先がルートノードとなるツリーを用いてフレームを転送することで、宛先への最低コストによるフレーム伝送のほか、リンク利用率の向上、さらにはルートノード障害への耐性強化を行う。

## 【 0 4 9 2 】

図 2 0 を参照すると、本発明の第 4 の実施の形態は、第 1 の実施の形態における図 8 において、ツリーマネージャ 1 1 5 1 が、ネットワークに存在するノードの数だけ設置されている点において異なる。

## 【 0 4 9 3 】

ツリーマネージャ 1 1 5 1 は、本発明の第 1 の実施の形態における図 8 に示すツリーマネージャ 1 1 5 1 と同様の機能を有する。

## 【 0 4 9 4 】

ツリーマネージャ 1 1 5 2 および、ツリーマネージャ 1 1 5 3 は、ツリーマネージャ 1 1 5 1 と同様のツリーマネージャである。以後、ツリーマネージャ 1 1 5 1 ～ツリーマネージャ 1 1 5 3 を代表してツリーマネージャ 1 1 5 1 を用いて記述を行うが、ツリーマネージャ 1 1 5 1 についての記述は、特に断りのない限りツリーマネージャ 1 1 5 2 ～ツリーマネージャ 1 1 5 3 においても同様に適用可能である。

## 【 0 4 9 5 】

ツリーマネージャはツリーセクタ 1 1 6 によって、ネットワーク内に存在するノードの数、もしくは、サブネット等を区切り階層化している場合は、同一階層内に存在するノードの数だけ作成される。したがって、ツリーマネージャは、1 個以上無限大まで増える可能性があるが、図 2 0 においては前記ツリーマネージャを、ツリーマネージャ 1 1 5 1 ～1 1 5 3 としてまとめて表現する。

## 【 0 4 9 6 】



図 2 0 のツリーセクタ 1 1 6 は、本発明の第 1 の実施の形態における図 8 に示すツリーセクタ 1 1 6 の機能のほか、ネットワークもしくは階層内に新規ノードの存在を検出した場合に、新規のツリーマネージャを生成する機能を有する。また、前記新規ノードの検出を、他のノードに対して通知する機能、および、他のノードから新規ノードの検出通知を受信し、ツリーマネージャを生成する機能を有する。さらに、ノードの削除を検出して、ツリーマネージャを削除する機能、前記削除ノードの検出を他のノードに対して通知する機能、および、他のノードからノードの削除通知を受信し、ツリーマネージャを削除する機能を有する。

## 【 0 4 9 7 】

図 2 1 は、本実施の形態の図 2 0 における、タグをキーとして出力ポートを決定する、フォワーディングテーブル 1 1 4 の構成例である。

## 【 0 4 9 8 】

タグフィールド 1 1 4 1 は、サーチのインデックスになるフィールドであり、このフィールドの情報と、受信したフレームのタグに書かれている内容が一致するかどうかを調べる。

## 【 0 4 9 9 】

出力ポート 1 1 4 2 は、受信したフレームのタグに書かれている内容とタグにフィールド 1 1 4 1 の内容が一致した場合に、前記フレームをどのポートに転送すべきか記述するフィールドである。

## 【 0 5 0 0 】

なお、本実施の形態は、本動作例に示すような、タグの内容によって転送先ポートを決定するタグフォワーディングを行う場合のみならず、従来より行われている、MAC アドレスによって転送先を決定する通常の MAC アドレス転送においても、同様に適用可能である。この場合は、出力ポートフィールド 1 1 4 2 に、対応する MAC アドレスに対応する複数のポートが記載される場合もある。

## 【 0 5 0 1 】

図 2 2 は、本実施の形態の図 2 0 における、宛先 MAC アドレスをキーとして挿入するタグを決定する、タグテーブル 1 1 7 の構成例である。

## 【 0 5 0 2 】

宛先MACアドレス1171は、サーチのインデックスになるフィールドであり、このフィールドの情報と、受信したフレームの宛先MACアドレスフィールド、すなわち、MAC DAフィールドに書かれている内容が一致するかどうか調べ、一致した場合には、前記受信フレームに対して、挿入タグフィールド1172に記載のタグを挿入する。

## 【 0 5 0 3 】

挿入タグフィールド1172は、宛先MACアドレスフィールド1171に対する、挿入すべきタグが記載されるフィールドである。本実施例では、宛先ノードIDが記載されており、このIDが、タグとしてフレームに挿入される。

## 【 0 5 0 4 】

図23は、ノード11をルートノードとするスパニングツリーの構成図である、ツリー61の構成図である。ツリー61は、ノード11のプライオリティ値を、ノード12～ノード16の各ノードよりも小さい値に設定して作成される。ツリー61は、ノード11に向かうフレームの送信および、ノード11よりノード12～ノード16の各ノードに対して、ブロードキャストフレームを送信する場合に利用される。

## 【 0 5 0 5 】

図24は、ノード12をルートノードとするスパニングツリーの構成図である、ツリー62の構成図である。ツリー62は、ノード12のプライオリティ値を、ノード11およびノード13～ノード16の各ノードよりも小さい値に設定して作成される。ツリー62は、ノード12に向かうフレームの送信および、ノード12よりノード11およびノード13～ノード16の各ノードに対して、ブロードキャストフレームを送信する場合に利用される。

## 【 0 5 0 6 】

図25は、ノード13をルートノードとするスパニングツリーの構成図である、ツリー63の構成図である。ツリー63は、ノード13のプライオリティ値を、ノード11～ノード12およびノード14～ノード16の各ノードよりも小さい値に設定して作成される。ツリー63は、ノード13に向かうフレームの送信

および、ノード 1 3 よりノード 1 1～ノード 1 2 およびノード 1 4～ノード 1 6 の各ノードに対して、ブロードキャストフレームを送信する場合に利用される。

#### 【 0 5 0 7 】

図 2 6 は、ノード 1 4 をルートノードとするスパニングツリーの構成図である、ツリー 6 4 の構成図である。ツリー 6 4 は、ノード 1 4 のプライオリティ値を、ノード 1 1～ノード 1 3 およびノード 1 5～ノード 1 6 の各ノードよりも小さい値に設定して作成される。ツリー 6 4 は、ノード 1 4 に向かうフレームの送信および、ノード 1 4 よりノード 1 1～ノード 1 3 およびノード 1 5～ノード 1 6 の各ノードに対して、ブロードキャストフレームを送信する場合に利用される。

#### 【 0 5 0 8 】

図 2 7 は、ノード 1 5 をルートノードとするスパニングツリーの構成図である、ツリー 6 5 の構成図である。ツリー 6 5 は、ノード 1 5 のプライオリティ値を、ノード 1 1～ノード 1 4 およびノード 1 6 の各ノードよりも小さい値に設定して作成される。ツリー 6 5 は、ノード 1 5 に向かうフレームの送信および、ノード 1 5 よりノード 1 1～ノード 1 4 およびノード 1 6 の各ノードに対して、ブロードキャストフレームを送信する場合に利用される。

#### 【 0 5 0 9 】

図 2 8 は、ノード 1 6 をルートノードとするスパニングツリーの構成図である、ツリー 6 6 の構成図である。ツリー 6 6 は、ノード 1 6 のプライオリティ値を、ノード 1 1～ノード 1 5 よりも小さい値に設定して作成される。ツリー 6 6 は、ノード 1 6 に向かうフレームの送信および、ノード 1 6 よりノード 1 1～ノード 1 5 の各ノードに対して、ブロードキャストフレームを送信する場合に利用される。

#### 【 0 5 1 0 】

次に、図 2 3～図 2 8 を利用して、ノード 1 3 が、ノード 1 1～1 2 およびノード 1 4～1 6 ですでに構成されているネットワークに、新たに追加された場合の、ツリー 6 3 の作成動作について説明する。

#### 【 0 5 1 1 】

前記ネットワークにノード 1 3 が追加されると、ノード 1 3 は隣接ノードより

送信される B P D U フレームを受信して、新たに検出した識別タグごとにツリーマネージャを生成する。この例では、ノード 1 1 ~ 1 2 およびノード 1 4 ~ 1 6 の各ノードをルートノードとする、5 個のツリーマネージャが生成される。

#### 【 0 5 1 2 】

次に、ノード 1 3 は、ノード I D よりタグ I D を生成し、自ノードのプライオリティ値を低く設定したツリーマネージャを生成し、そのツリーマネージャから出力される B P D U フレームに対して、前記タグ I D を付加して送信する。ここで、タグ I D は 4 3 であったとする。

#### 【 0 5 1 3 】

ノード 1 2 およびノード 1 6 は、タグ I D が 4 3 の B P D U を新たに受信し、ツリーマネージャを生成してから、タグ I D 4 3 を付加した B P D U を隣接ノードに対して送信する。

#### 【 0 5 1 4 】

以上の B P D U 送信動作が繰り返されることにより、ツリー 6 3 が完成する。

#### 【 0 5 1 5 】

次に、図 2 3 ~ 図 2 8 を参照して、前記の各図におけるノード 1 1 ~ ノード 1 6 の各ノードが、ノード 1 1 ~ ノード 1 6 の各ノードにフレームを送信する場合の手順について述べ、送信したフレームが最低コスト経路で宛先へ届けられること、並びに、リンク資源の負荷分散が行われていることを示す。なお、各リンクのコストは等しく、すでに各図におけるツリー 6 1 ~ ツリー 6 6 の各ツリーは構成が完了し、トポロジが安定しているとする。

#### 【 0 5 1 6 】

ノード 1 2 ~ ノード 1 6 の各ノードから、ノード 1 1 へフレームを送信する場合は、ツリー 6 1 を使用する。例えば、ノード 1 5 からノード 1 1 へフレームを送信する場合は、ノード 1 5 は、データフレームにツリー 6 1 の識別タグであるタグ I D 4 1 を付加して送信する。

#### 【 0 5 1 7 】

ノード 1 1 およびノード 1 3 ~ ノード 1 6 の各ノードから、ノード 1 2 へフレームを送信する場合は、ツリー 6 2 を使用する。例えば、ノード 1 4 からノード

1 2 ヘフレームを送信する場合は、ノード 1 4 は、データフレームにツリー 6 2 の識別タグであるタグ I D 4 2 を付加して送信する。

## 【 0 5 1 8 】

ノード 1 1 ～ノード 1 2 およびノード 1 4 ～ノード 1 6 の各ノードから、ノード 1 3 ヘフレームを送信する場合は、ツリー 6 3 を使用する。例えば、ノード 1 1 からノード 1 3 ヘフレームを送信する場合は、ノード 1 1 は、データフレームにツリー 6 3 の識別タグであるタグ I D 4 3 を付加して送信する。

## 【 0 5 1 9 】

ノード 1 1 ～ノード 1 3 およびノード 1 5 ～ノード 1 6 の各ノードから、ノード 1 4 ヘフレームを送信する場合は、ツリー 6 4 を使用する。例えば、ノード 1 2 からノード 1 4 ヘフレームを送信する場合は、ノード 1 2 は、データフレームにツリー 6 4 の識別タグであるタグ I D 4 4 を付加して送信する。

## 【 0 5 2 0 】

ノード 1 1 ～ノード 1 4 およびノード 1 6 の各ノードから、ノード 1 5 ヘフレームを送信する場合は、ツリー 6 5 を使用する。例えば、ノード 1 6 からノード 1 5 ヘフレームを送信する場合は、ノード 1 6 は、データフレームにツリー 6 5 の識別タグであるタグ I D 4 5 を付加して送信する。

## 【 0 5 2 1 】

ノード 1 1 ～ノード 1 5 の各ノードから、ノード 1 6 ヘフレームを送信する場合は、ツリー 6 6 を使用する。例えば、ノード 1 4 からノード 1 6 ヘフレームを送信する場合は、ノード 1 4 は、データフレームにツリー 6 6 の識別タグであるタグ I D 4 6 を付加して送信する。

## 【 0 5 2 2 】

ノード 1 1 が、フレームをネットワーク内の全ノードにブロードキャスト送信する場合は、ツリー 6 1 を使用する。例えば、ノード 1 1 は、宛先がブロードキャストとなるデータフレームに対して、ツリー 6 1 の識別タグであるタグ I D 4 1 を付加して送信する。

## 【 0 5 2 3 】

ノード 1 2 が、フレームをネットワーク内の全ノードにブロードキャスト送信

する場合は、ツリー 6 2 を使用する。例えば、ノード 1 2 は、宛先がブロードキャストとなるデータフレームに対して、ツリー 6 2 の識別タグであるタグ ID 4 2 を付加して送信する。

【 0 5 2 4 】

ノード 1 3 が、フレームをネットワーク内の全ノードにブロードキャスト送信する場合は、ツリー 6 3 を使用する。例えば、ノード 1 3 は、宛先がブロードキャストとなるデータフレームに対して、ツリー 6 3 の識別タグであるタグ ID 4 3 を付加して送信する。

【 0 5 2 5 】

ノード 1 4 が、フレームをネットワーク内の全ノードにブロードキャスト送信する場合は、ツリー 6 4 を使用する。例えば、ノード 1 4 は、宛先がブロードキャストとなるデータフレームに対して、ツリー 6 4 の識別タグであるタグ ID 4 4 を付加して送信する。

【 0 5 2 6 】

ノード 1 5 が、フレームをネットワーク内の全ノードにブロードキャスト送信する場合は、ツリー 6 5 を使用する。例えば、ノード 1 5 は、宛先がブロードキャストとなるデータフレームに対して、ツリー 6 5 の識別タグであるタグ ID 4 5 を付加して送信する。

【 0 5 2 7 】

ノード 1 6 が、フレームをネットワーク内の全ノードにブロードキャスト送信する場合は、ツリー 6 6 を使用する。例えば、ノード 1 6 は、宛先がブロードキャストとなるデータフレームに対して、ツリー 6 6 の識別タグであるタグ ID 4 6 を付加して送信する。

【 0 5 2 8 】

以上に示す方法で送信時タグを付加してデータフレームを転送することで、前記データフレームを最低コスト経路で転送することができる。また、ルートノードが異なる複数のツリーを利用してフレームを転送するため、従来技術 1 および従来技術 2 に示したスパニングツリーのように、ルートノード付近にトラフィックが集中し、ルートノードから遠ざかるほどリンクの使用率が減るという現象が



発生せず、トラフィックの付加を分散できることがわかる。

【 0 5 2 9 】

次に、図 2 3 ～ 図 2 8 において、ノードに障害が発生した場合の動作について、ノード 1 2 に障害が発生した場合を例にして説明する。なお、最初の状態では、ツリー 6 1 ～ ツリー 6 6 は、すでに構築されて安定しているとする。

【 0 5 3 0 】

ツリー 6 1 は、ノード 1 2 が障害によって停止すると、IEEE 8 0 2 . 1 w で規定されるラピッドスパニングツリー方式により、ノード 1 3 からノード 1 1 へのルートとして、リンク 2 5、リンク 2 7、リンク 2 6、リンク 2 3 を経由するルートが選ばれ、ノード 1 1 へのフレーム及び、ノード 1 1 から各ノードへブロードキャストされるフレームの転送が継続される。

【 0 5 3 1 】

ツリー 6 2 は、ノード 1 2 が障害によって停止すると、ノード 1 2 がルートノードであるので、ツリーを再構成しなくてはならない。ノード 1 2 以外の別のノードがルートノードとなり、ノード 1 2 の回復前にツリー 6 2 の再構成が行われる。この再構成には、IEEE 8 0 2 . 1 D では数十秒、IEEE 8 0 2 . 1 w では数秒間かかるが、そもそもツリー 6 2 は、各ノードからルート 1 2 へ向けて送信されるフレーム、および、ルート 1 2 から各ノードに向けてブロードキャスト送信されるフレームのためのツリーなので、再構成に時間がかかっても、ノード 1 2 以外の他のノード間で行われる通信には影響を与えない。

【 0 5 3 2 】

ツリー 6 3 は、ノード 1 2 が障害によって停止すると、IEEE 8 0 2 . 1 w で規定されるラピッドスパニングツリー方式により、ノード 1 1 からノード 1 3 へのルートとして、リンク 2 3、リンク 2 6、リンク 2 7、リンク 2 5 を経由するルートが選ばれ、ノード 1 3 へのフレーム及び、ノード 1 3 から各ノードへブロードキャストされるフレームの転送が継続される。

【 0 5 3 3 】

ツリー 6 4 は、ノード 1 2 が障害によって停止すると、IEEE 8 0 2 . 1 w で規定されるラピッドスパニングツリー方式により、ツリーが再構成され、各ノ

ードからノード 1 4 へ送信されるフレーム及び、ノード 1 4 から各ノードへブロードキャストされるフレームの転送が継続される。

【 0 5 3 4 】

ツリー 6 5 は、ノード 1 2 が障害によって停止すると、I E E E 8 0 2 . 1 w で規定されるラピッドスパニングツリー方式により、ツリーが再構成され、各ノードからノード 1 5 へのフレーム及び、ノード 1 5 から各ノードへブロードキャストされるフレームの転送が継続される。

【 0 5 3 5 】

ツリー 6 6 は、ノード 1 2 が障害によって停止すると、I E E E 8 0 2 . 1 w で規定されるラピッドスパニングツリー方式により、ノード 1 1 からノード 1 6 へのルートとして、リンク 2 3、リンク 2 6、リンク 2 7 を経由するルートが選ばれ、ノード 1 6 へのフレーム及び、ノード 1 6 から各ノードへブロードキャストされるフレームの転送が継続される。

【 0 5 3 6 】

次に、図 2 9 と図 3 0 を参照して、第 1 の実施の形態の図 7 におけるクライアントの一部が、デュアルホーミングによって、複数のノードと接続されている場合の、スパニングツリーの構成方法について述べる。

【 0 5 3 7 】

図 2 9 において、クライアント 9 7 は、1 つ以上のクライアントの集合であり、リンク 8 7 およびリンク 8 8 を通じて、ノード 1 1 およびノード 1 6 との間でフレームの送受信を行う機能を有する。

【 0 5 3 8 】

リンク 8 7 は、クライアント 9 7 からノード 1 5、及び、ノード 1 5 からクライアント 9 7 を結ぶ双方向リンクである。

【 0 5 3 9 】

リンク 8 8 は、クライアント 9 7 からノード 1 6、及び、ノード 1 6 からクライアント 9 7 を結ぶ双方向リンクである。

【 0 5 4 0 】

図 2 9 におけるクライアント 9 7 のように、複数のノードと接続されたクライ

アント群が存在する場合は、前記クライアントを仮想的なノードとみなして、スパニングツリーの設定を行う。

【 0 5 4 1 】

図 3 0 は、図 2 9 において、クライアント 9 7 を仮想ノード 1 8 と見立てた場合のネットワーク構成図である。

【 0 5 4 2 】

スパニングツリー 7 4 は、ノード 1 8 をルートノードとするスパニングツリーである。ノード 1 1 ～ノード 1 6 の各ノードから、ノード 1 8 へ送信されたフレームは、スパニングツリー 7 4 を利用してノード 1 8、つまりクライアント 9 7 に到着する。また、クライアント 9 7、すなわちノード 1 8 から送信されたブロードキャストフレームも、スパニングツリー 7 4 に沿って、ノード 1 1 ～ノード 1 6 の各ノードにブロードキャストされる。

【 0 5 4 3 】

なお、ノード 1 8 は仮想的なノードであり、ノード 1 8 の実際の動作は、ノード 1 5 もしくは 1 6 が代行して行う。ノード 1 5 とノード 1 6 のどちらがノード 1 8 の動作を代行するかは、設定インタフェース 1 1 8 による手動設定のほか、ノード ID の小さい方に自動設定、ノード ID の大きい方に自動設定する方法で決定される。

【 0 5 4 4 】

次に、図 2 9 を参照して、クライアント 9 7 が、リンク 8 7 およびリンク 8 8 を通じて、ノード 1 5 およびノード 1 6 とデュアルホーミング接続されている場合において、仮想ノードを設置することなく通信を行う方法について説明する。

【 0 5 4 5 】

図 2 9 において、ノード 1 5 およびノード 1 6 は、設定インタフェースでの設定、もしくは、ラーニングによって、クライアント 9 7 が複数のノードに接続されていることを検知する。ノード 1 5 は、ノード 1 6 にクライアント 9 7 が接続されていることを検知する。ノード 1 6 は、ノード 1 5 にクライアント 9 7 が接続されていることを検知する。

【 0 5 4 6 】

ノード 1 5 および ノード 1 6 は、相互で制御メッセージの交換を行い、クライアント 9 7 へのフレームを、ノード 1 5 もしくは ノード 1 6 のどちらが転送するかを決定する。この転送ノードは、ノード ID の小さいノード、ノード ID の大きいノード、もしくは、設定で定められたノード等に決定される。

## 【 0 5 4 7 】

転送ノードが決定すると、クライアント 9 7 はノード 1 6 のみに接続されているとみなされ、フレーム転送が開始される。ノード 1 1 ~ ノード 1 6 は、クライアント 9 7 がノード 1 6 に接続されていると学習等で認識し、クライアント 9 7 宛のフレームを、ノード 1 6 をルートノードとするツリーの識別タグを付加して送信する。

## 【 0 5 4 8 】

ノード 1 5 および ノード 1 6 は、常に Keep Alive 等により相互に状況を監視しあう。もし、ノード 1 5 がノード 1 6 の動作を確認できないときは、ノード 1 5 はクライアント 9 7 からのフレームを、ノード 1 1 ~ 1 6 に転送するようになる。すると、ノード 1 1 ~ ノード 1 6 は、クライアント 9 7 から送信されたフレームに、ノード ID として 1 5 が付加されていることを学習し、クライアント 9 7 宛のフレームをノード 1 5 に向けて送信するようになる。

## 【 0 5 4 9 】

以上の動作により、クライアント 9 7 は、フレームを送受信することができる。以上の動作は、ノード 1 5 とノード 1 6 が入れ替わった場合においても、同様に適用可能である。

## 【 0 5 5 0 】

次に、図 2 9 を参照して、クライアント 9 7 が、リンク 8 7 およびリンク 8 8 を通じて、ノード 1 5 および ノード 1 6 とデュアルホーミング接続されている場合において、仮想ノードを設置することなく通信を行い、また、障害検出ノードがネットワーク内の全ノードに対して切替通知を送信することで、クライアント 9 7 の接続先が変更されたことを高速に通知する方法について説明する。

## 【 0 5 5 1 】

図 2 9 において、ノード 1 5 および ノード 1 6 は、設定インタフェース 1 1 8

での設定、もしくは、ラーニングによって、クライアント 9 7 が複数のノードに接続されていることを検知する。ノード 1 5 は、ノード 1 6 にクライアント 9 7 が接続されていることを検知する。ノード 1 6 は、ノード 1 5 にクライアント 9 7 が接続されていることを検知する。

#### 【 0 5 5 2 】

ノード 1 5 およびノード 1 6 は、相互で制御メッセージの交換を行い、クライアント 9 7 へのフレームを、ノード 1 5 もしくはノード 1 6 のどちらが転送するかを決定する。この転送ノードは、ノード ID の小さいノード、ノード ID の大きいノード、もしくは、設定で定められたノード等に決定される。

#### 【 0 5 5 3 】

転送ノードが決定すると、クライアント 9 7 はノード 1 6 のみに接続されているとみなされ、フレーム転送が開始される。ノード 1 1 ～ノード 1 6 は、クライアント 9 7 がノード 1 6 に接続されていると学習等で認識し、クライアント 9 7 宛のフレームを、ノード 1 6 をルートノードとするツリーの識別タグを付加して送信する。

#### 【 0 5 5 4 】

ノード 1 5 およびノード 1 6 は、常に Keep Alive 等により相互に状況を監視しあう。もし、ノード 1 5 がノード 1 6 の動作を確認できないときは、ノード 1 5 はクライアント 9 7 からのフレームを、ノード 1 1 ～1 6 に転送するようになる。さらに、ノード 1 5 は、クライアント 9 7 宛のフレーム転送を、ノード 1 6 に代わってノード 1 5 が受け持つようになったことを、ネットワーク内の全ノードに通知する。

#### 【 0 5 5 5 】

ノード 1 1 ～ノード 1 6 は、前記通知を受け取り、クライアント 9 7 宛てのフレームに対してノード 1 5 宛のタグを挿入し、クライアント 9 7 宛のフレームをノード 1 5 に向けて送信するようになる。

#### 【 0 5 5 6 】

以上の動作により、クライアント 9 7 は、フレームを送受信することができる。以上の動作は、ノード 1 5 とノード 1 6 が入れ替わった場合においても、同様

に適用可能である。

【 0 5 5 7 】

次に、本実施の形態の効果について説明する。

【 0 5 5 8 】

従来、宛先への最低コスト経路が必ずしも選択されとは限らなかった。

【 0 5 5 9 】

本実施の形態では、宛先がルートノードとなるツリーを用いてフレームを転送することにより、宛先への最低コスト経路を選択することができる。

【 0 5 6 0 】

また従来、リンク利用率が低い一方、ルートノード付近に負荷が集中するという問題があった。

【 0 5 6 1 】

本実施の形態では、ルートノードの異なるスパニングツリーを複数系統設定することにより、リンク利用率を高め、ルートノード付近に負荷を集中させずに分散させることが可能である。

【 0 5 6 2 】

さらに従来、ルートノード障害時のツリー構築に時間がかかり、その間ネットワークが停止するという問題があった。

【 0 5 6 3 】

本実施の形態では、宛先がルートノードとなるツリーを用いてフレームを転送することにより、ルートノードが宛先となるフレーム以外のフレームが、ルートノード障害の影響により長時間転送不能となることがないため、ルートノード障害によるネットワーク停止を回避可能である。

【 0 5 6 4 】

またこの結果、輻輳発生の可能性を下げる事が可能である。

【 0 5 6 5 】

(第 5 の実施の形態)

以下、本発明の第 5 の実施の形態について、図面を参照して詳細に説明する。

【 0 5 6 6 】



本発明の第 5 の実施の形態は、第 1 の実施の形態において、B P D U のバージョンを識別し、低速な I E E E 8 0 2 . 1 D すなわち従来技術 1 を用いている区間についてはコストを大きく設定し、高速な I E E E 8 0 2 . 1 w すなわち従来技術 2 を用いている区間についてはコストを小さく設定したスパニングツリーを生成する場合に対応する。

## 【 0 5 6 7 】

I E E E 8 0 2 . 1 D を利用する区間は、障害時のルート切り替えが遅く、スパニングツリーの再構成にも時間がかかるため、この区間を通過するようなツリーを設定すると、障害発生時やルート変更に時間がかかり、輻輳が発生してフレームが欠落する問題があった。

## 【 0 5 6 8 】

本実施の形態においては、I E E E 8 0 2 . 1 D 利用区間のコストを大きく設定することで、スパニングツリーが I E E E 8 0 2 . 1 D 利用区間を通過して設定されるのを防ぎ、障害時の切替やルート変更を高速化し、輻輳の発生やフレームの欠落を防止する。

## 【 0 5 6 9 】

図 3 1 を参照すると、本発明の第 5 の実施の形態は、第 1 の実施の形態における図 1 0 において、コスト操作器 1 1 5 1 6 が追加されている点において異なる。

## 【 0 5 7 0 】

ツリーコントローラ 1 1 5 1 4 は、本発明の第 1 の実施の形態におけるツリーコントローラ 1 1 5 1 4 の動作のほかに、受信した B P D U のバージョンを判別し、もし設定されたバージョンより低いバージョンの B P D U が到着した場合は、前記 B P D U を送信したノードが接続されているリンクのコストを、コスト操作器 1 1 5 1 6 によって再設定し、ツリーテーブル 1 1 5 1 5 に書き込む。なおこの動作は、ツリーセクタ 1 1 6 よりコスト変更の通知を受信するごとに、1 度のみ行う。

## 【 0 5 7 1 】

コスト操作器 1 1 5 1 6 は、ツリーコントローラ 1 1 5 1 4 より入力された値

に、あらかじめ設定された値を加算し、ツリーコントローラ 1 1 5 1 4 に返答する。

【 0 5 7 2 】

ツリーコントローラ 1 1 5 1 4 は、B P D U 送受信機 1 1 5 1 2 より B P D U 受信通知を受けると、通知の内容に従ってツリーテーブル 1 1 5 1 5 に値を設定していく。P D U 受信通知には、受信した B P D U のバージョンおよび受信ポートに関する情報も含まれ、前記情報もツリーテーブル 1 1 5 1 5 において保持される。

【 0 5 7 3 】

ツリーコントローラ 1 1 5 1 4 は、ツリーセクタ 1 1 6 より、コスト情報が通知されると、通知された情報に従ってテーブルにコスト値を設定する。このとき、あらかじめ設定されたバージョンより古いバージョンの B P D U を受信したポートに対してコストを設定する場合は、ツリーセクタ 1 1 6 より通知されたコストをコスト操作器 1 1 5 1 6 に通知する。

【 0 5 7 4 】

コスト操作器 1 1 5 1 6 は、ツリーコントローラ 1 1 5 1 4 より入力された値に、あらかじめ設定された値を加算し、ツリーコントローラ 1 1 5 1 4 に返答する。

【 0 5 7 5 】

ツリーコントローラ 1 1 5 1 4 は、コスト操作器 1 1 5 1 6 より返答されたコストを、該当ポートのコストとして、ツリーテーブル 1 1 5 1 5 に通知する。

【 0 5 7 6 】

ツリーコントローラ 1 1 5 1 4 は、すべてのポートに対するコスト更新が完了した後で、スパニングツリーアルゴリズムにしたがってツリーを再構成する。

【 0 5 7 7 】

次に、図 3 2 ～図 3 4 を参照して、本実施の形態におけるスパニングツリー作成動作について述べる。

【 0 5 7 8 】

図 3 2 ～図 3 4 において、ノード 1 2 が I E E E 8 0 2 . 1 w 非対応のノード

であり、IEEE 802.1Dにのみ対応するとする。ノード12以外のノード、すなわち、ノード11、ノード13、ノード14、ノード15およびノード16は、IEEE 802.1wに対応するとする。

#### 【0579】

ノード11、ノード15、ノード13の各ノードは、ノード12から送信されるBPDUフレーム内のバージョン情報もしくはプロトコルIDにより、ノード12が802.1Dのみに対応するノードであると認識する。

#### 【0580】

ノード11、ノード15、ノード13の各ノードは、リンク21、リンク22およびリンク24の各リンクのコストを、他のリンクのコストよりも十分大きく設定する。ここでは、リンク21、リンク22、リンク24のコストが10に設定され、他のリンク、すなわち、リンク23、リンク26、リンク27、リンク25のコストが1に設定されたとする。

#### 【0581】

図32は、前記コスト設定の状態において、ノード11もしくはノード14がルートノードとなる場合の、スパニングツリーの構成図である。

#### 【0582】

図33は、前記コスト設定の状態において、ノード15、ノード16もしくはノード14がルートノードとなる場合の、スパニングツリーの構成図である。

#### 【0583】

図34は、前記コスト設定の状態において、ノード13もしくはノード16がルートノードとなる場合の、スパニングツリーの構成図である。

#### 【0584】

図32～図34に示すように、本実施の形態では、障害回復に時間がかかるIEEE 802.1D利用区間を避けるようにしてツリーを構成することが可能であり、障害時にネットワーク全体が受ける影響を小さくし、高速な障害回復ができる。

#### 【0585】

次に、本実施の形態の効果について説明する。

【 0 5 8 6 】

従来、I E E E 8 0 2 . 1 D を利用する区間は、障害時のルート切り替えが遅く、スパニングツリーの再構成にも時間がかかった。

【 0 5 8 7 】

本実施の形態では、I E E E 8 0 2 . 1 D 利用区間のコストを大きく設定することにより、スパニングツリーがI E E E 8 0 2 . 1 D 利用区間を通過して設定されるのを防ぎ、障害時の切替やルート変更を高速化し、輻輳発生の可能性やフレームの欠落の可能性を低下させることができる。

【 0 5 8 8 】

(第 6 の実施の形態)

以下、本発明の第 6 の実施の形態について、図面を参照して詳細に説明する。

【 0 5 8 9 】

本発明の第 6 の実施の形態は、第 1 の実施の形態において、分別器が B P D U のバージョンを識別し、ツリーセクタを通じて、ツリーマネージャを低速な I E E E 8 0 2 . 1 D 利用区間の数だけ作成することで、I E E E 8 0 2 . 1 D 利用区間で障害が発生した場合に、前記区間を迂回する経路を高速に提供する場合に対応する。

【 0 5 9 0 】

I E E E 8 0 2 . 1 D を利用する区間は、障害時のルート切り替えが遅く、スパニングツリーの再構成にも時間がかかるため、この区間において障害が発生すると、障害発生時やルート変更に時間がかかり、輻輳が発生してフレームが欠落する問題があった。

【 0 5 9 1 】

本実施の形態においては、I E E E 8 0 2 . 1 D 利用区間の数だけツリーマネージャを作成し、相異なる 1 つの I E E E 8 0 2 . 1 D 利用区間のコストを大きく設定したツリーを I E E E 8 0 2 . 1 D 利用区間分作成し、障害等により前記区間を迂回する必要が生じた場合は、前記区間に対して大きなコストを振ったツリーを利用することで、迂回操作を高速に行い、輻輳の発生やフレームの欠落を防止できる。

## 【 0 5 9 2 】

図 3 5 を参照すると、本発明の第 6 の実施の形態は、本発明の第 1 の実施の形態における図 8 において、ツリーマネージャ 1 1 5 1 が I E E E 8 0 2 . 1 D 利用区間の数だけ存在する点において異なる。

## 【 0 5 9 3 】

ツリーマネージャ 1 1 5 1 は、本発明の第 1 の実施の形態における図 8 に示すツリーマネージャ 1 1 5 1 の機能のほか、受信した B P D U フレームが、バージョンフィールドもしくはその他の手段により I E E E 8 0 2 . 1 D に準拠した B P D U フレームであると確認できた場合は、ツリーセクタ 1 1 6 に対して、前記 B P D U フレームに対して 8 0 2 . 1 D フレーム受信通知を送信する。8 0 2 . 1 D フレーム受信通知には、8 0 2 . 1 D 準拠の B P D U フレームを送信したノードのノード I D が記入される。

## 【 0 5 9 4 】

ツリーマネージャ 1 1 5 2 および、ツリーマネージャ 1 1 5 3 は、ツリーマネージャ 1 1 5 1 と同様のツリーマネージャである。以後、ツリーマネージャ 1 1 5 1 ～ツリーマネージャ 1 1 5 3 を代表してツリーマネージャ 1 1 5 1 を用いて記述を行うが、ツリーマネージャ 1 1 5 1 についての記述は、特に断りのない限りツリーマネージャ 1 1 5 2 ～ツリーマネージャ 1 1 5 3 においても同様に適用可能である。

## 【 0 5 9 5 】

ツリーマネージャはツリーセクタ 1 1 6 によって I E E E 8 0 2 . 1 D 利用区間の数だけ作成される。したがって、ツリーマネージャは、1 個以上無限大まで増える可能性があるが、図 3 5 においては前記ツリーマネージャを、ツリーマネージャ 1 1 5 1 ～1 1 5 3 としてまとめて表現する。

## 【 0 5 9 6 】

ツリーセクタ 1 1 6 は、本発明の第 1 の実施の形態における図 8 に示すツリーセクタ 1 1 6 の機能のほか、ツリーマネージャ 1 1 5 1 ～ツリーマネージャ 1 1 5 3 の何れかのツリーマネージャより 8 0 2 . 1 D フレーム受信通知を受けた場合は、新規のツリーマネージャを生成する機能、8 0 2 . 1 1 D 利用ノード

をネットワーク内の他のノードに通知する機能、および、他のノードより送信された、802.1D利用ノードの通知に基づき、ツリーマネージャを生成する機能を有する。

## 【0597】

ツリーセクタ116は、前記機能のほか、802.1D利用ノードがバージョンアップ等の何らかの原因により802.1wが利用可能になったことを検知し、ツリーマネージャを削除する機能、そして、前記削除に関する情報を、ネットワーク内の他のノードに伝達する機能、および、他のノードより通知された前記削除情報に基づき、ツリーマネージャを削除する機能を有する。

## 【0598】

図36は、ノード11をルートノードとし、IEEE802.1wの通常の手順によって作成される、ツリー67の構成図である。

## 【0599】

図37は、ノード11をルートノードとし、ツリー21のコストを大きくして作成される、ツリー68の構成図である。このツリーは、リンク21に障害が発生した際にも利用される。

## 【0600】

図38は、ノード11をルートノードとし、ツリー22のコストを大きくして作成される、ツリー69の構成図である。このツリーは、リンク22に障害が発生した際にも利用される。

## 【0601】

図39は、ノード11をルートノードとし、ツリー24のコストを大きくして作成される、ツリー70の構成図である。このツリーは、リンク24に障害が発生した際にも利用される。

## 【0602】

次に、図36～図39を参照して、図36～図39におけるノード12がIEEE802.1w未対応のノードであり、リンク21とリンク22とリンク24において、IEEE802.1Dが利用されている場合の動作について述べる。なお、ルートノードは、ノード11とする。



## 【 0 6 0 3 】

まず、IEEE 8 0 2 . 1 w に従った通常の手順によって、図 3 6 に示すスパニングツリー 6 7 が形成される。このとき、ノード 1 2 は 8 0 2 . 1 w 未対応のノードであるため、ノード 1 2 からは、IEEE 8 0 2 . 1 D のプロトコル ID が付加された B P D U フレームが送信される。

## 【 0 6 0 4 】

ノード 1 1 は、ノード 1 2 より IEEE 8 0 2 . 1 D のプロトコル ID が付加された B P D U を受信すると、新規のツリーマネージャを生成して、前記ツリーマネージャに対して、リンク ID およびノード ID 等から算出する固有のタグ ID を割り当て、G V R P フレームもしくはその他のフレームによって、全ノードに対して新規グループの作成をブロードキャスト通知する。ここでは新規のタグ ID として、タグ ID 4 8 が割り当てられたとする。このとき、リンク 2 1 のコストは大きく設定する。

## 【 0 6 0 5 】

ノード 1 2 ～ノード 1 6 は、ノード 1 1 より送信された新規グループ作成通知を受信転送し、ツリーマネージャを生成して B P D U の交換を開始する。新規に作成されたツリーマネージャの間で交換される B P D U には、タグ ID 4 8 のタグが付加される。ここで作成されるスパニングツリーを、ツリー 6 8 とする。

## 【 0 6 0 6 】

ノード 1 3 は、ノード 1 2 より IEEE 8 0 2 . 1 D のプロトコル ID が付加された B P D U を受信すると、新規のツリーマネージャを生成して、前記ツリーマネージャに対して、リンク ID およびノード ID 等から算出する固有のタグ ID を割り当て、G V R P フレームもしくはその他のフレームによって、全ノードに対して新規グループの作成をブロードキャスト通知する。ここでは新規のタグ ID として、タグ ID 4 9 が割り当てられたとする。このとき、リンク 2 2 のコストは大きく設定する。

## 【 0 6 0 7 】

ノード 1 1 ～ノード 1 2 およびノード 1 4 ～ノード 1 6 は、ノード 1 3 より送信された新規グループ作成通知を受信転送し、ツリーマネージャを生成して B P

D Uの交換を開始する。新規に作成されたツリーマネージャの間で交換される B P D Uには、タグ I D 4 9のタグが付加される。ここで作成されるスパニングツリーを、ツリー 6 9とする。

## 【 0 6 0 8 】

ノード 1 5は、ノード 1 2より I E E E 8 0 2. 1 Dのプロトコル I Dが付加された B P D Uを受信し、新規のツリーマネージャを生成して、前記ツリーマネージャに対して、リンク I Dおよびノード I D等から算出する固有のタグ I Dを割り当て、G V R Pフレームもしくはその他のフレームによって、全ノードに対して新規グループの作成をブロードキャスト通知する。ここでは新規のタグ I Dとして、タグ I D 5 0が割り当てられたとする。このとき、リンク 2 4のコストは大きく設定する。

## 【 0 6 0 9 】

ノード 1 1～ノード 1 4およびノード 1 6は、ノード 1 5より送信された新規グループ作成通知を受信転送し、ツリーマネージャを生成して B P D Uの交換を開始する。新規に作成されたツリーマネージャの間で交換される B P D Uには、タグ I D 5 0のタグが付加される。ここで作成されるスパニングツリーを、ツリー 7 0とする。

## 【 0 6 1 0 】

平常時においては、各ノード間の通信はツリー 6 7を用いて行われ、ツリー 6 8～ツリー 7 0は利用されない。

## 【 0 6 1 1 】

ここで、もしリンク 2 1に障害が発生したとすると、ノード 1 1は、リンク 2 1の障害を検知して、直ちに転送に使用するツリーを、ツリー 6 7からツリー 6 8に切り替える。また、利用タググループ変更通知を全ノードに対してブロードキャスト送信し、転送に使用するタグをタグ I D 4 8に切り替えるように通知する。

## 【 0 6 1 2 】

ノード 1 2～ノード 1 6の各ノードは、ノード 1 1より送信された利用タググループ変更通知を受信し、自ノードから送信するフレームに対してタグ I D 4 8

のタグを挿入し、転送に使用するツリーを、ツリー 6 7 からツリー 6 8 に切り替える。

## 【 0 6 1 3 】

なお、リンク 2 1 に障害が発生した場合の、ツリー 6 8 への切替は、ノード 1 1 ではなくノード 1 2 が行ってもかまわない。もしノード 1 2 がリンク 2 1 の障害を検出した場合は、直ちに転送に使用するツリーを、ツリー 6 7 からツリー 6 8 に切り替える。また、利用タググループ変更通知を全ノードに対してブロードキャスト送信し、転送に使用するタグをタグ I D 4 8 に切り替えるように通知する。以降の動作は、ノード 1 1 が障害を検出した場合と同様である。

## 【 0 6 1 4 】

ツリー 6 7 は、リンク 2 1 の障害によって再構成されるが、リンク 2 1 では 8 0 2 . 1 D に従った再構成が行われるために、再構成が完了するまでに時間がかかる可能性がある。

## 【 0 6 1 5 】

本実施の形態では、リンク 2 1 の障害時に、転送に利用されるツリーを、ツリー 6 7 からツリー 6 8 にただちに切り替えることで、ツリー 6 7 の再構成を待たずに、フレームの転送を継続することができる。

## 【 0 6 1 6 】

以上、リンク 2 1 に障害が発生した場合の動作について述べたが、前記動作はリンク 2 2 もしくはリンク 2 4 に障害が発生した場合においても、同様に適用可能である。

## 【 0 6 1 7 】

次に、本実施の形態の効果について説明する。

## 【 0 6 1 8 】

従来、I E E E 8 0 2 . 1 D を利用する区間は、障害時のルート切り替えが遅く、スパニングツリーの再構成にも時間がかかった。

## 【 0 6 1 9 】

本実施の形態では、I E E E 8 0 2 . 1 D 利用区間の数だけツリーマネージャを作成し、相異なる 1 つの I E E E 8 0 2 . 1 D 利用区間のコストを大きく設定

したツリーを I E E E 8 0 2 . 1 D 利用区間分作成し、障害等により前記区間を迂回する必要がある場合は、前記区間に対して大きなコストを振ったツリーを利用するよう切り替えることで、迂回操作を高速に行い、輻輳発生の可能性やフレームの欠落の可能性を低下させることができる。

【 0 6 2 0 】

(第 7 の実施の形態)

次いで、本発明の第 7 の実施の形態について、図面を参照して詳細に説明する。

【 0 6 2 1 】

本発明の第 7 の実施の形態は、第 1 の実施の形態において、障害検出器を用いて定期的に短い間隔で障害検出用フレームを送信し、前記障害検出用フレームが到着しなくなったことにより、障害検出を行い、障害情報をリソースモニタおよびツリーセクタを通じて、ツリーマネージャに通知する点が異なる。

【 0 6 2 2 】

I E E E 8 0 2 . 1 D および I E E E 8 0 2 . 1 w では、定期的に送信される H E L L O フレームの未着により、障害を検出していた。しかしながら、H E L L O フレームの送信間隔は長いため、障害を検出するまでに長い時間がかかった。

【 0 6 2 3 】

本実施の形態では、障害検出器から定期的に短い間隔で障害検出用フレームを送信し、前記障害検出用フレームの一定個数もしくは一定時間以上の未着により、高速な障害検出を行う。

【 0 6 2 4 】

図 4 0 を参照すると、本発明の第 7 の実施の形態は、第 1 の実施の形態における図 4 において、障害検出用フレームを送受信して障害を検出する、障害検出器 1 2 0 が追加されている点において異なる。

【 0 6 2 5 】

障害検出器 1 2 0 は、障害検出用フレームをフレーム転送器 1 1 1 を通じて隣接ノードに対して定期的に送信し、また、隣接ノードより送信された障害検出用

フレームを、フレーム転送器 1 1 1 より受信し、隣接ノードから送られる障害検出用フレームが一定時間以上受信できない場合、および、一定個数以上の障害検出用フレームの未着を検出した場合に、リソースモニタ 1 1 9 に対して障害検出通知を送信する。

## 【 0 6 2 6 】

リソースモニタ 1 1 9 は、第 1 の実施の形態におけるリソースモニタ 1 1 9 の機能のほか、障害検出器 1 2 0 より障害通知を受信し、前記障害通知をツリーセクタ 1 1 6 に転送する機能を有する。

## 【 0 6 2 7 】

ツリーセクタ 1 1 6 は、第 1 の実施の形態におけるツリーセクタ 1 1 6 の機能のほか、リソースモニタ 1 1 9 より障害通知を受信し、これをツリーマネージャ 1 1 5 1 および 1 1 5 2 に転送する機能を有する。

## 【 0 6 2 8 】

ツリーマネージャ 1 1 5 1 および 1 1 5 2 は、本発明の第 1 の実施の形態におけるツリーマネージャ 1 1 5 1 および 1 1 5 2 の機能のほか、ツリーセクタ 1 1 6 より障害通知を受信し、IEEE 8 0 2 . 1 w もしくは IEEE 8 0 2 . 1 D に従い、スパニングツリーの再構成を行う。

## 【 0 6 2 9 】

次に、図 4 0 を参照して、本実施の形態においてノード 1 1 がリンク 2 1 の障害を検知する場合の動作例について述べる。

## 【 0 6 3 0 】

障害検出器 1 2 0 は、定期的に、フレーム転送器 1 1 1 およびリンク 2 1 もしくはリンク 2 4 を経由して、隣接ノードであるノード 1 2 およびノード 1 5 に障害検出用フレームを送信する。

## 【 0 6 3 1 】

障害検出器 1 2 0 は、また、隣接しているノード 1 2 およびノード 1 5 より、リンク 2 1 もしくはリンク 2 4 と、フレーム転送器 1 1 1 を経由して送信された障害検出用フレームを受信する。このとき、障害検出器 1 2 0 は、前記障害検出用フレームが到着したポートの ID も、識別することができる。

## 【 0 6 3 2 】

障害検出器 1 2 0 は、障害検出用フレームが到着すると、前記障害検出用フレームが到着したポートのタイマを動作させ、一定時間が経過すると、通知を発信するように設定する。

## 【 0 6 3 3 】

障害検出器 1 2 0 は、障害検出用フレームを受信すると、ポートごとに設置されたタイマをリセットする。例えば、リンク 2 1 より障害検出用フレームを受信した場合は、リンク 2 1 が接続されているポートのタイマがリセットされる。また、リンク 2 4 より障害検出用フレームを受信した場合は、リンク 2 4 が接続されているポートのタイマがリセットされる。

## 【 0 6 3 4 】

ここで、障害検出器 1 2 0 において、もしリンク障害等により、障害検出用フレームが一定時間以上到着しない場合は、タイマがリセットされないためタイムアウトが起こる。障害検出器 1 2 0 は、タイムアウトが発生すると、何らかの障害が発生していると認識し、リソースモニタ 1 1 9 およびツリーセクタ 1 1.6 を通じて、ツリーマネージャ 1 1 5 1 およびツリーマネージャ 1 1 5 2 に将棋ア発生を通知する。

## 【 0 6 3 5 】

通知を受けたツリーマネージャ 1 1 5 1 およびツリーマネージャ 1 1 5 2 は、障害発生ポートが使用不可能なものとして、直ちにスパニングツリーを再構成し、障害回避を行う。

## 【 0 6 3 6 】

次に、本実施の形態の効果について説明する。

## 【 0 6 3 7 】

従来、スパニングツリープロトコルで利用されている H E L L O フレームの送信間隔が長かったため、高速な障害検出ができなかった。

## 【 0 6 3 8 】

本実施の形態では、短い間隔で障害検出用フレームを送受信する障害検出器を追加することにより、H E L L O フレームよりも高速な障害検出が可能である。



【 0 6 3 9 】

また、この結果、輻輳発生およびフレーム欠落の可能性を下げる事が可能である。

【 0 6 4 0 】

(実施の形態 8)

以下、本発明の第 8 の実施の形態について、図面を参照して詳細に説明する。

【 0 6 4 1 】

本発明の第 8 の実施の形態は、第 1 の実施の形態において、宛先ノード別に利用するタグおよびスパニングツリーを切り替え、さらに、宛先のノードをルートノードに設定することができる構成となっている。

【 0 6 4 2 】

IEEE 802.1D および IEEE 802.1w が動作しているネットワークにおいてフレームを送信した場合、宛先への最低コスト経路が必ずしも選択されるとは限らず、使用されないリンクが出現するほか、ルートノードに負荷が集中する、ルートノード障害時に長時間ネットワークが停止するなどの問題があった。

【 0 6 4 3 】

本実施の形態においては、宛先がルートノードとなるツリーを用いてフレームを転送することで、宛先への最低コストによるフレーム伝送のほか、リンク利用率の向上、さらにはルートノード障害への耐性強化を実現する。

【 0 6 4 4 】

図 4 1 を参照すると、本発明の第 8 の実施の形態は、第 1 の実施の形態における図 8 において、フレーム転送器 1 1 1 がフレーム転送器 1 1 1 γ に、フォワーディングテーブル 1 1 4 がフォワーディングテーブル 1 1 4 γ になり、さらにツリーマネージャ 1 1 5 1 が、動作を変更した上でネットワークに存在するノードの数だけ設置され、ツリーマネージャ 1 1 5 1 γ ~ 1 1 5 3 γ になっている点において異なる。

【 0 6 4 5 】

図 4 1 においては、本実施の形態を図 2 3 に示すノード 1 1 に適用した場合を

示している。

【 0 6 4 6 】

フレーム転送器 1 1 1 γ は、リンク 2 1 もしくはリンク 2 3、およびタグ挿入器 1 1 2 より受信したフレームを、フォワーディングテーブル 1 1 4 γ の出力ポート 1 1 4 2 の記述に従い、リンク 2 1 もしくはリンク 2 3、および、タグ削除器 1 1 3 もしくはツリーセクタ 1 1 6 に転送する。この際、もし出力ポート 1 1 4 2 の記述が初期値である場合は、前記受信フレームを廃棄する。

【 0 6 4 7 】

フォワーディングテーブル 1 1 4 γ は、フォワーディングテーブル 1 1 4 と同様のフォワーディングテーブルである。

【 0 6 4 8 】

ツリーマネージャ 1 1 5 1 γ は、ツリーマネージャ 1 1 5 1 と構成は同じであるが、その機能と動作において異なっている。以後、ツリーマネージャ 1 1 5 1 γ ～ ツリーマネージャ 1 1 5 3 γ を代表して、ツリーマネージャ 1 1 5 1 γ を用いて記述を行うが、ツリーマネージャ 1 1 5 1 γ についての記述は、特に断りのない限りツリーマネージャ 1 1 5 2 γ ～ ツリーマネージャ 1 1 5 3 γ においても同様に適用可能である。

【 0 6 4 9 】

図 4 2 は、フォワーディングタグをキーとして出力ポートを決定する、フォワーディングテーブル 1 1 4 γ の構成例である。

【 0 6 5 0 】

タグフィールド 1 1 4 1 は、フォワーディングテーブル 1 1 4 内のタグフィールドと同様のタグフィールドである。

【 0 6 5 1 】

出力ポート 1 1 4 2 は、フォワーディングテーブル 1 1 4 内の出力ポートと同様の出力ポートであり、ユニキャストフレームの転送時にフレーム転送器 1 1 1 γ によって参照される。

【 0 6 5 2 】

図 4 2 において、出力ポート 1 1 4 2 の「END」は、エッジノードにおいて

宛先ノードIDに対する出力ポートエントリが自ノードであることを示す識別子であり、例えば自ノードIDと等しいノード識別子エントリに対して出力ポートフィールドに記述される。

#### 【0653】

図43は、ツリーマネージャ1151γの構成を示す、ブロック図である。

#### 【0654】

図23に示すようにスパニングツリーが構成されている場合、各ノードのポートは、そのスパニングツリーの構成に応じて、図54のようにルートポート(Root Port)、ディジグネイテッドポート(Designated Port)又はオルタネートポート(Alternate Port)として決定される。図54においては、ルートポートを(R)、ディジグネイテッドポートを(D)、オルタネートポートを(ALT)として示している。もちろん、上記ポートの種別は、スパニングツリーの構成(ルートノードの位置)によって変化する。

#### 【0655】

ツリーマネージャ1151γが送受信するBPDUフレームに、タググループ41のタグが付加されている場合、ツリーコントローラ11514γは、IEEE802.1wもしくはIEEE802.1Dプロトコルによりルートポート(Root Port)を決定する。本実施の形態においてはフォワーディングテーブル114γの、タググループ41のエントリに、出力ポート1142として決定したルートポートを設定する。この場合、最低6個のBPDUフレームが送受信される。

#### 【0656】

ここで、もしルートポートが存在しない場合(図54のノード11)には、出力ポート1142の欄は、自ノード宛ポートに設定し、フレーム転送器に入力されたタググループ41のタグつきフレームを、タグ削除器113に転送する。

#### 【0657】

次に、図23および図44を用いて、ユニキャストフレーム転送動作について、ノード13からノード11へのユニキャストフレーム転送を例に説明する。

図44は、スパニングツリー61における各ノードのポート設定およびフォー

ディングテーブルの設定の状況示す表である。なお、この図 4 4 は、実際のテーブル内容を示すものではなく、あくまで設定状況を示すものである。

#### 【 0 6 5 8 】

図 2 3 に示す状態において、タググループ 4 1 のツリーであるツリー 6 1 は、すでに構築されて安定しており、ノード 1 1 ~ ノード 1 6 の各ノードにおけるルートポートは、図 4 4 におけるルートポート 6 1 0 2 に示すように決定されており、この結果より、各ノードのフォワーディングテーブル 1 1 4 における出力ポートは、図 4 4 における出力ポート 1 1 4 2 に示すように決定されているものとする。

#### 【 0 6 5 9 】

ここで、ツリー 6 1 は、ノード 1 1 をルートノードとするツリーを表している。また、タグ 4 1 は、ツリー 6 1 を示すタグの ID (値) を表している。すなわち、タグ 4 1 をフレームに付加するということは、ツリー 6 1 を用いてフレーム転送することを意味する。このように、ツリーの番号とノードの番号とタグの番号とのそれぞれは、一の位の数字がそれぞれ対応することを示している。例えばツリー 6 2 は、ノード 1 2 をルートとするツリーを示しており、タグ 4 2 は、ツリー 6 2 を示すタグの ID (値) を表している、この点は、以下の説明においても同様である。

#### 【 0 6 6 0 】

また、ここでは一例として上述のように、ツリー、ノード、タグにそれぞれ異なる番号を付した場合について説明をしたが、これ以外に、それぞれ対応するツリー、ノード、タグに同一の番号 (ID) を付して、それぞれの対応関係を単純明確化して管理することも可能である。

#### 【 0 6 6 1 】

まず、ノード 1 3 がノード 1 1 宛のユニキャストフレームに、タググループ 4 1 のタグを付加してフレームを送信する。この際、ノード 1 3 におけるタググループ 4 1 のフレームの出力先ポートは、ツリー 6 1 のルートポートであるリンク 2 2 側のポートが指定されている。したがって、前記フレームはリンク 2 2 側に出力される。

## 【 0 6 6 2 】

ノード 1 2 は、リンク 2 2 よりフレームを受信すると、タググループ 4 1 をキーとしてフォワーディングテーブルを検索し、出力ポートとしてリンク 2 1 側のポートを得る。そして、リンク 2 1 側に前記受信フレームを出力する。

## 【 0 6 6 3 】

ノード 1 1 は、リンク 2 1 よりフレームを受信すると、自ノード宛であることを確認し、タグ削除器 1 1 3 にフレームを転送する。

## 【 0 6 6 4 】

以上の動作により、タググループ 4 1 のタグおよびスパニングツリー 6 1 を用いて、最低コスト経路によって、ノード 1 3 からノード 1 1 へのユニキャストフレームを転送することが可能となる。

## 【 0 6 6 5 】

次に、上述した第 8 の実施の形態の効果について説明する。

## 【 0 6 6 6 】

従来、宛先への最低コスト経路が必ずしも選択されとは限らなかったが、本実施の形態では、宛先がルートノードとなるツリーを用いてフレームを転送することにより、宛先への最低コスト経路を選択することができる。

## 【 0 6 6 7 】

また従来、リンク利用率が低い一方、ルートノード付近に負荷が集中するという問題があったが、本実施の形態では、ルートノードの異なるスパニングツリーを複数系統設定することにより、リンク利用率を高め、ルートノード付近に負荷を集中させずに分散させることが可能である。

## 【 0 6 6 8 】

さらに従来、ルートノード障害時のツリー構築に時間がかかり、その間ネットワークが停止するという問題があったが、本実施の形態では、宛先がルートノードとなるツリーを用いてフレームを転送することにより、ルートノードが宛先となるフレーム以外のフレームが、ルートノード障害の影響により長時間転送不能となることがないため、ルートノード障害によるネットワーク停止を回避可能である。また、この結果、輻輳発生の可能性を下げる事が可能である。

## (実施の形態 9)

以下、本発明の第 9 の実施の形態について、図面を参照して詳細に説明する。

## 【 0 6 6 9 】

本発明の第 9 の実施の形態は、第 8 の実施の形態において、フォワーディングテーブルに、通常のユニキャスト用出力ポートのほかに、複数のブロードキャスト用出力ポートを記載し、ブロードキャストフレームを転送することができる構成となっている。

## 【 0 6 7 0 】

本実施の形態においては、ブロードキャストフレームの送信元ノードがルートノードとなるツリーを用いてブロードキャストフレームを転送することで、ブロードキャストフレームを各ノードに最短経路で送信することができ、高速な転送が可能になる。

## 【 0 6 7 1 】

図 4 5 を参照すると、本発明の第 9 の実施の形態は、第 8 の実施の形態における図 4 1 において、フレーム転送器 1 1 1 がフレーム転送器 1 1 1  $\beta$  に、フォワーディングテーブル 1 1 4 がフォワーディングテーブル 1 1 4  $\beta$  に、ツリーマネージャ 1 1 5 1 ~ 1 1 5 3 がツリーマネージャ 1 1 5 1  $\beta$  ~ 1 1 5 3  $\beta$  になっている点において異なる。

## 【 0 6 7 2 】

図 4 5 においては、本実施の形態を図 2 3 に示すノード 1 1 に適用した場合を示している。

## 【 0 6 7 3 】

フレーム転送器 1 1 1  $\beta$  は、リンク 2 1 もしくはリンク 2 4、およびタグ挿入器 1 1 2 より受信したフレームを、フォワーディングテーブル 1 1 4  $\beta$  の記述に従い、リンク 2 1 もしくはリンク 2 4、および、タグ削除器 1 1 3 もしくはツリーセクタ 1 1 6 に転送する。

## 【 0 6 7 4 】

この際、もし入力されたフレームがユニキャストフレームである場合は、前記受信フレームをフォワーディングテーブル 1 1 4  $\beta$  の出力ポート 1 1 4 2 に記述



されたポートに転送する。

#### 【 0 6 7 5 】

もし、入力されたフレームがブロードキャストフレームである場合は、フォーワーディングテーブル 1 1 4  $\beta$  のブロードキャスト出力ポート 1 1 4 4 に記述された複数のポートおよびタグ削除器に、前記受信フレームをコピーして転送する。さらにもし、ブロードキャスト出力ポートに初期値が設定されていた場合は、前記受信フレームをタグ削除器のみに転送する。

#### 【 0 6 7 6 】

ブロードキャストフレームとユニキャストフレームの判別は、送信先 MAC アドレス 3 2 0 1、もしくは拡張タグのプライオリティ 5 0 0 3、もしくは拡張タグ情報フィールド 5 0 0 4 によって行う。

#### 【 0 6 7 7 】

フォーワーディングテーブル 1 1 4  $\beta$  は、フォーワーディングテーブル 1 1 4 に、ブロードキャスト出力ポート 1 1 4 4 欄が付加されたフォーワーディングテーブルである。ブロードキャスト出力ポート 1 1 4 4 欄は、対応するタグ ID が示すノードから送信されたブロードキャスト信号を受信したときに、転送先となるポートを示している。本実施形態におけるフォーワーディングテーブルの一例を図 4 6 に示す。図 4 6 では、転送先ポートを示すものとして、リンク名を用いている。図 4 6 は、図 2 3 ～ 2 8 に記載の物理トポロジからなるネットワークにおけるノード 1 1 のフォーワーディングテーブル 1 1 4  $\beta$  を示している。上述の通り、本発明では、宛先ノード毎に、フレーム転送に用いるツリーが異なっている。例えば、ノード 1 1 宛のフレームは、図 2 3 のツリーを用い、ノード 1 2 宛のフレームは図 2 4 のツリーを用いてフレーム転送を行う。

#### 【 0 6 7 8 】

ここで、タグ 4 2 が付加されたフレームのブロードキャストの場合を例として以下に図 4 6 の見方を説明する。タグ 4 2 が付加されているということは、ノード 1 2 から発信されたブロードキャストフレームであることを示している。そして、ノード 1 2 から発信されたブロードキャストフレームは、本発明では、図 2 4 のツリーを用いて転送される。従って、ノード 1 1 では、このブロードキャス

トフレームを受信すると、リンク 2 3 側に転送する必要がある。このような考え方に基づいて、図 4 6 は作成される。

#### 【 0 6 7 9 】

ツリーマネージャ 1 1 5 1  $\beta$  は、ツリーマネージャ 1 1 5 1 と構成は同じであるが、その機能と動作において異なっている。以後、ツリーマネージャ 1 1 5 1  $\beta$  ～ツリーマネージャ 1 1 5 3  $\beta$  を代表して、ツリーマネージャ 1 1 5 1  $\beta$  を用いて記述を行うが、ツリーマネージャ 1 1 5 1  $\beta$  についての記述は、特に断りのない限りツリーマネージャ 1 1 5 2  $\beta$  ～ツリーマネージャ 1 1 5 3  $\beta$  においても同様に適用可能である。

#### 【 0 6 8 0 】

図 4 6 は、タグをキーとして出力ポートを決定する、フォワーディングテーブル 1 1 4  $\beta$  の構成例である。

#### 【 0 6 8 1 】

タグフィールド 1 1 4 1 は、フォワーディングテーブル 1 1 4 内のタグフィールドと同様のタグフィールドである。

#### 【 0 6 8 2 】

出力ポート 1 1 4 2 は、フォワーディングテーブル 1 1 4 内の出力ポートと同様の出力ポートであり、ユニキャストフレームの転送時にフレーム転送器 1 1 1  $\beta$  によって参照される。

#### 【 0 6 8 3 】

ブロードキャスト出力ポート 1 1 4 4 は、ブロードキャストフレームの転送時に、フレーム転送器 1 1 1  $\beta$  によって参照される出力ポートである。この欄には複数のポートが記載され、もし 2 つ以上のポートが記載された場合は、記載されたポートの数だけフレームをコピーしてから転送する。もし、フレーム転送器 1 1 1  $\beta$  によって参照されたときに、この欄に設定された値が初期値のままである場合は、フレーム転送器 1 1 1  $\beta$  は、フレームをタグ削除器のみに転送する。

#### 【 0 6 8 4 】

なお、図 4 6 中の (END) とあるのは、ノードがエッジノードである場合に識別子「END」が記述されることを示している。

## 【 0 6 8 5 】

図 4 7 は、ツリーマネージャ 1 1 5 1  $\beta$  の構成を示す、ブロック図である。

## 【 0 6 8 6 】

ツリーマネージャ 1 1 5 1  $\beta$  が送受信する B P D U フレームに、タググループ 4 1 のタグが付加されている場合、ツリーコントローラ 1 1 5 1 4  $\beta$  は、I E E E 8 0 2 . 1 w もしくは I E E E 8 0 2 . 1 D プロトコルによりルートポート ( R o o t P o r t ) およびディジグネイテッドポート ( D e s i g n a t e d P o r t ) を決定する。本実施の形態では、フォワーディングテーブル 1 1 4  $\beta$  の、タググループ 4 1 のエントリに、出力ポート 1 1 4 2 として決定したルートポートを、ブロードキャスト出力ポート 1 1 4 4 として、1 つまたは複数の決定したディジグネイテッドポートを、それぞれ設定する。

## 【 0 6 8 7 】

ここで、もしルートポートが存在しない場合には、出力ポート 1 1 4 2 の欄は、自ノード宛ポートに設定し、フレーム転送器に入力されたタググループ 4 1 のタグつきフレームを、タグ削除器 1 1 3 に転送する。

## 【 0 6 8 8 】

また、もしディジグネイテッドポートが 1 つも存在しない場合には、ブロードキャスト出力ポート 1 1 4 4 の欄は、テーブルの初期値に設定する。

## 【 0 6 8 9 】

次に、図 2 3 および図 4 8 を用いて、ブロードキャストフレーム転送動作について説明する。

## 【 0 6 9 0 】

図 4 8 は、スパニングツリー 6 1 における各ノードのポート設定およびフォワーディングテーブルの設定を示す表である。

## 【 0 6 9 1 】

最初の状態では、タググループ 4 1 のツリーであるツリー 6 1 は、すでに構築されて安定しており、ノード 1 1 ~ ノード 1 6 の各ノードにおけるルートポートおよびディジグネイテッドポートは、図 4 8 におけるルートポート 6 1 0 2 およびディジグネイテッドポート 6 1 0 4 に示すように決定されており、この結果よ

り、各ノードのフォワーディングテーブル 1 1 4  $\beta$  における出力ポートおよび予備出力ポートは、図 4 8 における出力ポート 1 1 4 2 およびブロードキャスト出力ポート 1 1 4 4 に示すように決定されているものとする。

## 【 0 6 9 2 】

まず、ノード 1 1 がブロードキャストフレームに、タググループ 4 1 のタグを付加して送信する。この際、ノード 1 1 におけるタググループ 4 1 のブロードキャストフレームの出力先ポートは、ツリー 6 1 のディジグネイテッドポートであるリンク 2 1 側およびリンク 2 3 側のポートが指定されている。したがって、前記フレームはリンク 2 1 側およびリンク 2 3 側、さらにタグ削除器宛に、コピーされてから出力される。

## 【 0 6 9 3 】

ノード 1 2 は、リンク 2 1 よりブロードキャストフレームを受信すると、タググループ 4 1 をキーとしてフォワーディングテーブルを検索し、ブロードキャスト出力ポートとしてリンク 2 2 側のポートを得る。そして、リンク 2 2 側およびタグ削除器に、前記受信ブロードキャストフレームを出力する。

## 【 0 6 9 4 】

ノード 1 3 は、リンク 2 2 よりブロードキャストフレームを受信すると、タググループ 4 1 をキーとしてフォワーディングテーブルを検索し、ブロードキャスト出力ポートとして初期値を得る。そして、タグ削除器に前記受信フレームを出力する。

## 【 0 6 9 5 】

ノード 1 4 は、リンク 2 3 よりブロードキャストフレームを受信すると、タググループ 4 1 をキーとしてフォワーディングテーブルを検索し、ブロードキャスト出力ポートとしてリンク 2 6 側のポートを得る。そして、リンク 2 6 側およびタグ削除器に、前記受信フレームを出力する。

## 【 0 6 9 6 】

ノード 1 5 は、リンク 2 6 よりブロードキャストフレームを受信すると、タググループ 4 1 をキーとしてフォワーディングテーブルを検索し、ブロードキャスト出力ポートとしてリンク 2 7 側のポートを得る。そして、リンク 2 7 側および

タグ削除器に、前記受信フレームを出力する。

【 0 6 9 7 】

ノード 1 6 は、リンク 2 7 よりブロードキャストフレームを受信すると、タググループ 4 1 をキーとしてフォーワーディングテーブルを検索し、ブロードキャスト出力ポートとして初期値を得る。そして、タグ削除器に前記受信フレームを出力する。

【 0 6 9 8 】

以上の動作により、ノード 1 1 から出力されたブロードキャストフレームは、最低コスト経路によって、ネットワーク上の各ノードに転送される。

【 0 6 9 9 】

次に、上述した第 9 の実施の形態の効果について説明する。

【 0 7 0 0 】

従来、ブロードキャスト時に、宛先への最低コスト経路が必ずしも選択されるとは限らなかったが、本実施の形態では、送信元ノードがルートノードとなるツリーを用いてブロードキャストフレームを転送することにより、全ノードへの最低コスト経路を選択してブロードキャストフレームを転送することができる。

【 0 7 0 1 】

また従来、リンク利用率が低い一方、ルートノード付近に負荷が集中するという問題があったが、本実施の形態では、ルートノードの異なるスパニングツリーを複数系統設定することにより、リンク利用率を高め、ルートノード付近に負荷を集中させずに分散させることが可能である。

【 0 7 0 2 】

さらに従来、ルートノード障害時のツリー構築に時間がかかり、その間ネットワークが停止するという問題があったが、本実施の形態では、送信元ノードがルートノードとなるツリーを用いてブロードキャストフレームを転送することにより、ルートノードが送信元ノードとなるフレーム以外のブロードキャストフレームが、ルートノード障害の影響により長時間転送不能となることがないため、ルートノード障害によるネットワーク停止を回避可能である。また、この結果、輻輳発生の可能性を下げる事が可能である。

【 0 7 0 3 】

(実施の形態 1 0)

以下、本発明の第 1 0 の実施の形態について、図面を参照して詳細に説明する。

【 0 7 0 4 】

本発明の第 1 0 の実施の形態は、第 8 の実施の形態において、フォワーディングテーブルに出力ポートを 2 つ記載し、1 つの出力ポートが障害等で利用できない場合に、もう一方のポートを利用できるようにし、さらに、第 7 の実施の形態における障害検出器を利用し、高速な障害検出を行うことができる構成となっている。

【 0 7 0 5 】

本実施の形態においては、宛先がルートノードとなるツリーを用いてユニキャストフレームを転送し、さらにスパニングツリーによって決定される代替出力ポートをあらかじめフォワーディングテーブルに登録することで、高速な障害回復を行う。

【 0 7 0 6 】

図 4 9 を参照すると、本発明の第 8 の実施の形態は、第 4 の実施の形態における図 4 1 において、フレーム転送器 1 1 1 がフレーム転送器 1 1 1  $\alpha$  に、フォワーディングテーブル 1 1 4 がフォワーディングテーブル 1 1 4  $\alpha$  に、ツリーマネージャ 1 1 5 1 ~ 1 1 5 3 がツリーマネージャ 1 1 5 1  $\alpha$  ~ 1 1 5 3  $\alpha$  になり、さらに、第 7 の実施の形態に記した障害検出器 1 2 0 が追加されている点において異なる。

【 0 7 0 7 】

図 4 9 においては、本実施の形態を図 2 3 に示すノード 1 1 に適用した場合を示している。

【 0 7 0 8 】

フレーム転送器 1 1 1  $\alpha$  は、リンク 2 1 もしくはリンク 2 3、およびタグ挿入器 1 1 2 より受信したフレームを、フォワーディングテーブル 1 1 4  $\alpha$  の記述に従い、リンク 2 1 もしくはリンク 2 3、および、タグ削除器 1 1 3 もしくはツリ



ーセクタ 1 1 6 に転送する。

【 0 7 0 9 】

この際、もしリソースモニタ 1 1 9 より、フォワーディングテーブル 1 1 4  $\alpha$  の出力ポート 1 1 4 2 に記述されたポートに障害があることを検知している場合は、前記受信フレームを予備出力ポート 1 1 4 3 に記述されたポートに転送する。もし、出力ポート 1 1 4 2 に記述されたポートの障害を検知しているが、予備出力ポートの記述が初期値（又は未設定）である場合は、前記受信フレームを廃棄する。

【 0 7 1 0 】

フォワーディングテーブル 1 1 4  $\alpha$  は、フォワーディングテーブル 1 1 4 に、予備出力ポート 1 1 4 3 欄が付加されたフォワーディングテーブルである。

【 0 7 1 1 】

ツリーマネージャ 1 1 5 1  $\alpha$  は、ツリーマネージャ 1 1 5 1 と構成は同じであるが、その機能と動作において異なっている。以後、ツリーマネージャ 1 1 5 1  $\alpha$  ～ツリーマネージャ 1 1 5 3  $\alpha$  を代表して、ツリーマネージャ 1 1 5 1  $\alpha$  を用いて記述を行うが、ツリーマネージャ 1 1 5 1  $\alpha$  についての記述は、特に断りのない限りツリーマネージャ 1 1 5 2  $\alpha$  ～ツリーマネージャ 1 1 5 3  $\alpha$  においても同様に適用可能である。

【 0 7 1 2 】

図 5 0 は、タグをキーとして出力ポートを決定する、図 2 3 におけるノード 1 2 のフォワーディングテーブル 1 1 4  $\alpha$  の構成例である。

【 0 7 1 3 】

タグフィールド 1 1 4 1 は、フォワーディングテーブル 1 1 4 内のタグフィールドと同様のタグフィールドである。

【 0 7 1 4 】

出力ポート 1 1 4 2 は、フォワーディングテーブル 1 1 4 内の出力ポートと同様の出力ポートである。

【 0 7 1 5 】

予備出力ポート 1 1 4 3 は、出力ポート 1 1 4 2 に記載されたポートが利用で

きなくなった場合に使用する出力先ポートを記載するためのフィールドである。  
出力ポート 1 1 4 2 に記載されたポートが利用不能であることをフレーム転送器  
1 1 1  $\alpha$  が検知した場合、フレーム転送器 1 1 1 は当該エントリの予備出力ポ  
ート 1 1 4 3 に記載されたポートにフレームを転送する。

## 【 0 7 1 6 】

図 5 1 は、ツリーマネージャ 1 1 5 1  $\alpha$  の構成を示す、ブロック図である。

## 【 0 7 1 7 】

ツリーマネージャ 1 1 5 1  $\alpha$  が送受信する B P D U フレームに、タググループ  
4 1 のタグが付加されている場合、ツリーコントローラ 1 1 5 1 4  $\alpha$  は、I E E  
E 8 0 2 . 1 w プロトコルによりルートポート ( R o o t P o r t ) およびオ  
ルタネートポート ( A l t e r n a t e P o r t ) を決定する。本実施の形態  
では、フォワーディングテーブル 1 1 4  $\alpha$  の、タググループ 4 1 のエントリに、  
出力ポート 1 1 4 2 として決定したルートポートを、予備出力ポート 1 1 4 3 と  
して決定したオルタネートポートを、それぞれ設定する。

## 【 0 7 1 8 】

ここで、もしルートポートが存在しない場合には、出力ポート 1 1 4 2 の欄は  
、自ノード宛先ポートに設定し、フレーム転送器に入力されたタググループ 4 1  
のタグつきフレームを、タグ削除器 1 1 3 に転送する。

## 【 0 7 1 9 】

また、もしオルタネートポートが存在しない場合には、予備出力ポート 1 1 4  
3 の欄は、テーブルの初期値に設定する。

## 【 0 7 2 0 】

次に、図 2 3 におよび図 5 2 を用いて、リンクに障害が発生した場合のユニキ  
ャストフレーム転送動作について、リンク 2 1 に障害が発生した場合を例にして  
説明する。

図 5 2 は、スパニングツリー 6 1 における各ノードのポート設定およびフォー  
ワーディングテーブルの設定を示す表である。

## 【 0 7 2 1 】

最初の状態では、タググループ 4 1 のツリーであるツリー 6 1 は、すでに構築

されて安定しており、ノード 1 1 ~ ノード 1 6 の各ノードにおけるルートポートおよびオルタネートポートは、図 5 2 におけるルートポート 6 1 0 2 およびオルタネートポート 6 1 0 3 に示すように決定されており、この結果より、各ノードのフォワーディングテーブル 1 1 4  $\alpha$  における出力ポートおよび予備出力ポートは、図 5 2 における出力ポート 1 1 4 2 および予備出力ポート 1 1 4 3 に示すように決定されているものとする。

#### 【 0 7 2 2 】

まず、ノード 1 3 がノード 1 1 宛のユニキャストフレームに、タググループ 4 1 のタグを付加してフレームを送信する。この際、ノード 1 3 におけるタググループ 4 1 のフレームの出力先ポートは、ツリー 6 1 のルートポートであるリンク 2 2 側のポートが指定されている。したがって、前記フレームはリンク 2 2 側に出力される。

#### 【 0 7 2 3 】

この状態で、リンク 2 1 において障害が発生したとする。

#### 【 0 7 2 4 】

ノード 1 2 は、リンク 2 2 よりフレームを受信すると、タググループ 4 1 をキーとしてフォワーディングテーブルを検索し、出力ポートとしてリンク 2 1 側のポート、予備出力ポートとしてリンク 2 4 側のポートを得る。そして、リンク 2 1 側に前記受信フレームを出力しようとするが、リソースモニタよりリンク 2 1 の障害検出情報を受信しているため、予備出力ポートであるリンク 2 4 側に前記受信フレームを出力する。

#### 【 0 7 2 5 】

ノード 1 5 は、リンク 2 4 よりフレームを受信すると、タググループ 4 1 をキーとしてフォワーディングテーブルを検索し、出力ポートとしてリンク 2 6 側のポート、予備出力ポートとしてリンク 2 4 側のポートを得る。そして、リンク 2 6 側に障害が発生していないことを確認し、リンク 2 6 側に前記受信フレームを出力する。

#### 【 0 7 2 6 】

ノード 1 4 は、リンク 2 6 よりフレームを受信すると、タググループ 4 1 をキ

ーとしてフォワーディングテーブルを検索し、出力ポートとしてリンク 2 3 側のポート、予備出力ポートとして初期値を得る。そして、リンク 2 3 側に障害が発生していないことを確認し、リンク 2 3 側に前記受信フレームを出力する。

【 0 7 2 7 】

ノード 1 1 は、リンク 2 3 よりフレームを受信すると、自ノード宛であることを確認し、タグ削除器 1 1 3 にフレームを転送する。

【 0 7 2 8 】

以上の動作により、リンク 2 1 において障害が発生した場合、あらかじめ設定されている予備出力ポートを参照し、ノード 1 3 からノード 1 1 へのユニキャストフレームを転送する場合のルートとして、リンク 2 2、リンク 2 3、リンク 2 4、リンク 2 6、リンク 2 3 を経由するルートがただちに選ばれ、ノード 1 1 へのフレーム転送が継続される。したがって高速な障害時の迂回が可能であり、ネットワークの輻輳を回避できる。

【 0 7 2 9 】

次に、上述した第 1 0 の実施の形態の効果について説明する。

【 0 7 3 0 】

従来、宛先への最低コスト経路が必ずしも選択されとは限らなかったが、本実施の形態では、宛先がルートノードとなるツリーを用いてユニキャストフレームを転送することにより、宛先への最低コスト経路を選択することができる。

【 0 7 3 1 】

また従来、リンク利用率が低い一方、ルートノード付近に負荷が集中するという問題があったが、本実施の形態では、ルートノードの異なるスパニングツリーを複数系統設定することにより、リンク利用率を高め、ルートノード付近に負荷を集中させずに分散させることが可能である。

【 0 7 3 2 】

さらに従来、ルートノード障害時のツリー構築に時間がかかり、その間ネットワークが停止するという問題があったが、本実施の形態では、宛先がルートノードとなるツリーを用いてフレームを転送することにより、ルートノードが宛先となるフレーム以外のフレームが、ルートノード障害の影響により長時間転送不能

となることがないため、ルートノード障害によるネットワーク停止を回避可能である。また、この結果、輻輳発生の可能性を下げることも可能である。

【 0 7 3 3 】

さらに従来、ルートポート側リンク障害時の出力先ポートの切替に時間がかかり、その間、フレーム転送が停止するという問題があったが、本実施の形態では、あらかじめ出力リンク障害時のための予備出力リンクをフォワーディングテーブルに設定しておくことで、ルートポート側リンク、すなわち出力リンクの障害時において、高速な経路変更が可能である。この結果、輻輳発生の可能性を下げることも可能である。

【 0 7 3 4 】

本発明のネットワークにおけるスパニングツリー構成ノードの構成要素である各手段の機能については、それをハードウェア的に実現することは勿論として、上記した各手段の機能を実行するスパニングツリー再構成プログラム（アプリケーションプログラム）950をコンピュータ処理装置のメモリにロードしてコンピュータ処理装置を制御することで実現することができる。このスパニングツリー再構成プログラム950は、磁気ディスク、半導体メモリその他の記録媒体に格納され、その記録媒体からコンピュータ処理装置にロードされ、コンピュータ処理装置の動作を制御することにより、上述した各機能を実現する。

【 0 7 3 5 】

以上好ましい実施の形態および実施例を挙げて本発明を説明したが、本発明は必ずしも上記実施の形態及び実施例に限定されるものではなく、その技術的思想の範囲内において様々に変形して実施することができる。

【 0 7 3 6 】

【発明の効果】

以上説明したように、本発明によれば、以下のような効果が達成される。

【 0 7 3 7 】

第1に、輻輳発生確率を下げ、輻輳によるフレームの到着遅れや欠落が発生する頻度を減らすことができる。その理由は、構成変更前のスパニングツリーを運用したまま、新規追加ノードを含めたスパニングツリーを生成し、新規スパニ

ングツリーの安定後に利用するスパニングツリーを切り替え、さらに、ルートノードの異なるスパニングツリーを複数系統設定するからである。

【 0 7 3 8 】

第 2 に、ネットワークを停止させずに、スパニングツリーに属するノードの追加および削除等のスパニングツリー再構成できる。その理由は、構成変更前のスパニングツリーを運用したまま、新規追加ノードを含めたスパニングツリーを生成し、新規スパニングツリーの安定後に利用するスパニングツリーを切り替えるからである。

【 0 7 3 9 】

第 3 に、トラフィックの負荷分散ができる。その理由は、リンクコストを、空き帯域やサーバ負荷等の動的情報によって計算するからである。

【 0 7 4 0 】

第 4 に、経路変更にとまなうスパニングツリー再構成のために、ネットワークを停止させることなく負荷分散ができる。その理由は、変更前のツリーを運用したまま、コスト変更後のツリーを生成し、新規ツリーの安定後に利用するツリーを切り替えるからである。

【 0 7 4 1 】

第 5 に、宛先への最低コスト経路を選択することができる。その理由は、宛先がルートノードとなるツリーを用いてフレームを転送するからである。

【 0 7 4 2 】

第 6 に、リンク利用率を高め、ルートノード付近に負荷を集中させずに分散可能である。その理由は、ルートノードの異なるスパニングツリーを複数系統設定するからである。

【 0 7 4 3 】

第 7 に、ルートノード障害によるネットワーク停止を回避可能である。その理由は、宛先がルートノードとなるツリーを用いてフレームを転送することにより、ルートノードが宛先となるフレーム以外のフレームが、ルートノード障害の影響により長時間転送不能となることがないからである。

【 0 7 4 4 】



第 8 に、スパニングツリーが I E E E 8 0 2 . 1 D 利用区間を通過して設定されるのを防ぎ、障害時の切替やルート変更を高速化し、輻輳発生の可能性やフレームの欠落の可能性を低下させることが可能である。その理由は、I E E E 8 0 2 . 1 D 利用区間のコストを大きく設定し、スパニングツリーが I E E E 8 0 2 . 1 D 利用区間を通過して設定されるのを防ぐからである。

## 【 0 7 4 5 】

第 9 に、迂回操作を高速に行い、輻輳の発生やフレームの欠落を防止できる。その理由は、I E E E 8 0 2 . 1 D 利用区間の数だけツリーマネージャを作成し、相異なる 1 つの I E E E 8 0 2 . 1 D 利用区間のコストを大きく設定したツリーを I E E E 8 0 2 . 1 D 利用区間分作成し、障害等により前記区間を迂回する必要が生じた場合は、前記区間に対して大きなコストを振ったツリーを利用するよう切り替えるからである。

## 【 0 7 4 6 】

第 1 0 に、従来、スパニングツリープロトコルで利用されている H E L L O フレームの送信間隔が長かったため、高速な障害検出ができなかったが、短い間隔で障害検出用フレームを送受信する障害検出器を追加することにより、H E L L O フレームよりも高速な障害検出が可能となる。また、この結果、輻輳発生およびフレーム欠落の可能性を下げることが可能となる。

## 【 0 7 4 7 】

第 1 1 に、従来、ブロードキャスト時に、宛先への最低コスト経路が必ずしも選択されとは限らなかったが、本発明では、送信元ノードがルートノードとなるツリーを用いてブロードキャストフレームを転送することにより、全ノードへの最低コスト経路を選択してブロードキャストフレームを転送することができる。

## 【 0 7 4 8 】

第 1 2 に、送信元ノードがルートノードとなるツリーを用いてブロードキャストフレームを転送することにより、ルートノードが送信元ノードとなるフレーム以外のブロードキャストフレームが、ルートノード障害の影響により長時間転送不能となることがないため、ルートノード障害によるネットワーク停止を回避可

能である。また、この結果、輻輳発生の可能性を下げる事が可能である。

【 0 7 4 9 】

第 1 3 に、従来、宛先への最低コスト経路が必ずしも選択されとは限らなかったが、宛先がルートノードとなるツリーを用いてユニキャストフレームを転送することにより、宛先への最低コスト経路を選択することができる。

【 0 7 5 0 】

第 1 4 に、従来、ルートポート側リンク障害時の出力先ポートの切替に時間がかかり、その間、フレーム転送が停止するという問題があったが、本発明では、あらかじめ出力リンク障害時のための予備出力リンクをフォワーディングテーブルに設定しておくことで、ルートポート側リンク、すなわち出力リンクの障害時において、高速な経路変更が可能である。この結果、輻輳発生の可能性を下げる事が可能である。

【図面の簡単な説明】

【図 1】 従来の V L A N タグ付きイーサネット(R)フレームの構成例を示す図である。

【図 2】 本発明の拡張タグ付きイーサネット(R)フレームの構成例を示す図である。

【図 3】 本発明の拡張タグ付きイーサネット(R)フレームの他の構成例を示す図である。

【図 4】 本発明の拡張タグ格納領域の構成例を示す図である。

【図 5】 本発明における C o n f i g u r a t i o n B P D U フレームのフレーム構成を示すフォーマット図である。

【図 6】 本発明における T o p o l o g y C h a n g e N o t i f i c a t i o n B P D U フレームのフレーム構成を示すフォーマット図である。

【図 7】 本発明の第 1 の実施の形態の構成を示すブロック図である。

【図 8】 本発明の第 1 の実施の形態における、ノード 1 1 の構成を示すブロック図である。

【図 9】 本発明の第 1 の実施の形態における、フォワーディングテーブル 1 1 4 の構成例を示す表である。

【図 1 0】 本発明の第 1 の実施の形態における、ツリーマネージャ 1 1 5 1 の構成を示すブロック図である。

【図 1 1】 本発明の第 1 の実施の形態における、ツリーセクタ 1 1 6 の構成を示すブロック図である。

【図 1 2】 本発明の第 1 の実施の形態における、メインコントローラ 1 1 6 4 の動作を示す流れ図である。

【図 1 3】 本発明の第 1 の実施の形態における、タグテーブル 1 1 7 の構成例を示す表である。

【図 1 4】 本発明の第 1 の実施の形態における、ノード 7 0 0 が追加される前  
のスパニングツリー 5 1 の構成を示すブロック図である。

【図 1 5】 本発明の第 1 の実施の形態における、ノード 7 0 0 が追加された後  
のスパニングツリー 5 2 の構成を示すブロック図である。

【図 1 6】 本発明の第 1 の実施の形態における、制御フレームのやりとり  
を示すシーケンス図である。

【図 1 7】 本発明の第 2 の実施の形態における、ツリーセクタ 1 1 6 の  
構成  
を示すブロック図である。

【図 1 8】 本発明の第 2 の実施の形態における、メインコントローラ 1 1  
6 4  
の動作を示す流れ図である。

【図 1 9】 本発明の第 3 の実施の形態における、メインコントローラ 1 1  
6 4 の動作を示す流れ図である。

【図 2 0】 本発明の第 4 の実施の形態における、ノード 1 1 の構成を示す  
ブロック図である。

【図 2 1】 本発明の第 4 の実施の形態における、フォワーディングテーブル 1 1 4 の構成例を示す表である。

【図 2 2】 本発明の第 4 の実施の形態における、タグテーブル 1 1 7 の構

成例を示す表である。

【図 2 3】 本発明の第 4 の実施の形態における、ツリー 6 1 の構成を示すブロック図である。

【図 2 4】 本発明の第 4 の実施の形態における、ツリー 6 2 の構成を示すブロック図である。

【図 2 5】 本発明の第 4 の実施の形態における、ツリー 6 3 の構成を示すブロック図である。

【図 2 6】 本発明の第 4 の実施の形態における、ツリー 6 4 の構成を示すブロック図である。

【図 2 7】 本発明の第 4 の実施の形態における、ツリー 6 5 の構成を示すブロック図である。

【図 2 8】 本発明の第 4 の実施の形態における、ツリー 6 6 の構成を示すブロック図である。

【図 2 9】 本発明の第 4 の実施の形態の構成を示すブロック図である。

【図 3 0】 本発明の第 4 の実施の形態における、ツリー 7 4 の構成を示すブロック図である。

【図 3 1】 本発明の第 5 の実施の形態における、ツリーマネージャ 1 1 5 1 の構成を示すブロック図である。

【図 3 2】 本発明の第 5 の実施の形態における、ツリー 7 1 の構成を示すブロック図である。

【図 3 3】 本発明の第 5 の実施の形態における、ツリー 7 2 の構成を示すブロック図である。

【図 3 4】 本発明の第 5 の実施の形態における、ツリー 7 3 の構成を示すブロック図である。

【図 3 5】 本発明の第 6 の実施の形態における、ノード 1 1 の構成を示すブロック図である。

【図 3 6】 本発明の第 6 の実施の形態における、ツリー 6 7 の構成を示すブロック図である。

【図 3 7】 本発明の第 6 の実施の形態における、ツリー 6 8 の構成を示す

ブロック図である。

【図 3 8】 本発明の第 6 の実施の形態における、ツリー 6 9 の構成を示すブロック図である。

【図 3 9】 本発明の第 6 の実施の形態における、ツリー 7 0 の構成を示すブロック図である。

【図 4 0】 本発明の第 7 の実施の形態における、ノード 1 1 の構成を示すブロック図である。

【図 4 1】 本発明の第 8 の実施の形態における、ノード 1 1 の構成を示すブロック図である。

【図 4 2】 本発明の第 8 の実施の形態における、フォワーディングテーブル 1 1 4  $\gamma$  の構成例を示す表である。

【図 4 3】 本発明の第 8 の実施の形態における、ツリーマネージャ 1 1 5 1  $\gamma$  の構成を示すブロック図である。

【図 4 4】 本発明の第 8 の実施の形態における、ツリー 6 1 の設定状況例を示す表である。

【図 4 5】 本発明の第 9 の実施の形態における、ノード 1 1 の構成を示すブロック図である。

【図 4 6】 本発明の第 9 の実施の形態における、フォワーディングテーブル 1 1 4  $\beta$  の構成例を示す表である。

【図 4 7】 本発明の第 9 の実施の形態における、ツリーマネージャ 1 1 5 1  $\beta$  の構成を示すブロック図である。

【図 4 8】 本発明の第 9 の実施の形態における、ツリー 6 1 の設定状況例を示す表である。

【図 4 9】 本発明の第 1 0 の実施の形態における、ノード 1 1 の構成を示すブロック図である。

【図 5 0】 本発明の第 1 0 の実施の形態における、フォワーディングテーブル 1 1 4  $\alpha$  の構成例を示す表である。

【図 5 1】 本発明の第 1 0 の実施の形態における、ツリーマネージャ 1 1 5 1  $\alpha$  の構成を示すブロック図である。

【図 5 2】 本発明の第 1 0 の実施の形態における、ツリー 6 1 の設定状況例を示す表である。

【図 5 3】 本発明における、拡張フレームの他の構成例を示す図である。

【図 5 4】 図 2 3 に示すスパニングツリーの構成における各ノードのポート状態を説明する図である。

【符号の説明】

- 1 1 ~ 1 8 ノード
- 2 1 ~ 3 0 双方向リンク
- 3 1 現用系 B P D U (タググループ 4 1)
- 3 2 予備系 B P D U (タググループ 4 2)
- 3 3 現用系 B P D U (タググループ 4 2)
- 3 4 予備系 B P D U (タググループ 4 1)
- 3 5 利用タググループ変更通知 (G V R P 等)
- 4 1 ~ 5 0 タググループ
- 5 1 ~ 5 2 スパニングツリー
- 6 1 ~ 7 3 スパニングツリー]
- 8 1 ~ 8 8 双方向リンク
- 9 1 ~ 9 7 クライアント
- 1 1 1 フレーム転送器
- 1 1 1  $\alpha$  フレーム転送器
- 1 1 1  $\beta$  フレーム転送器
- 1 1 1  $\gamma$  フレーム転送器
- 1 1 2 タグ挿入器
- 1 1 3 タグ削除器
- 1 1 4 フォワーディングテーブル
- 1 1 4  $\alpha$  フォワーディングテーブル
- 1 1 4  $\beta$  フォワーディングテーブル
- 1 1 4  $\gamma$  フォワーディングテーブル
- 1 1 6 ツリーセクタ

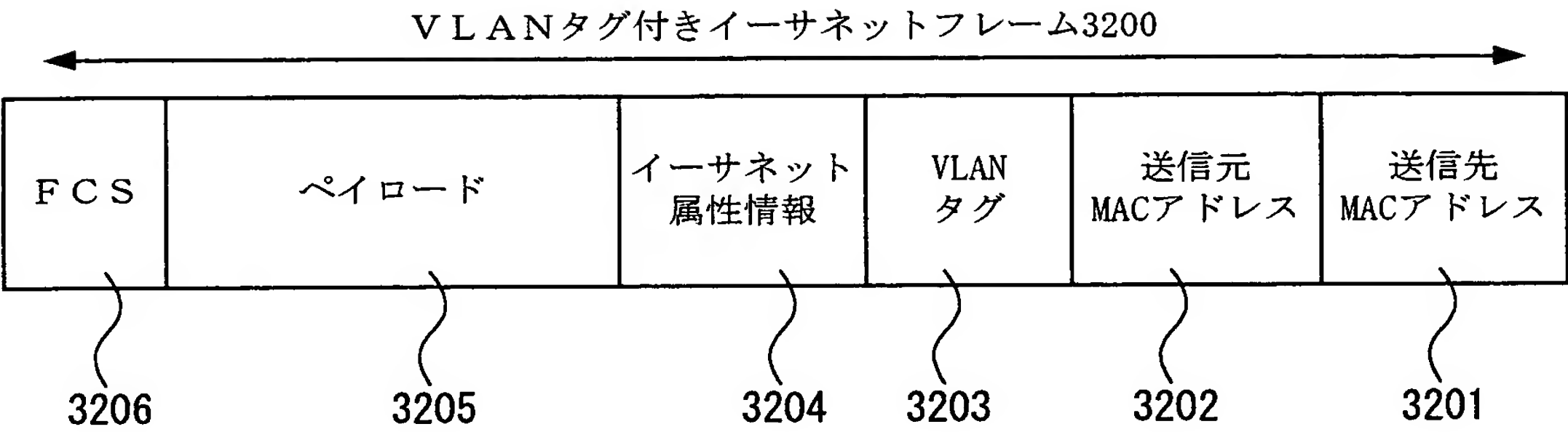


- 1 1 7 タグテーブル
- 1 1 8 設定インタフェース
- 1 1 9 リソースモニタ
- 1 1 4 3 予備出力ポート
- 1 1 4 4 ブロードキャスト出力ポート
- 1 1 5 0 分別器
- 1 1 5 1 ツリーマネージャ
- 1 1 5 1  $\alpha$  ツリーマネージャ
- 1 1 5 1  $\beta$  ツリーマネージャ
- 1 1 5 1  $\gamma$  ツリーマネージャ
- 1 1 5 2 ツリーマネージャ
- 1 1 5 3 ツリーマネージャ
- 1 1 6 1 タグ削除器
- 1 1 6 2 G V R P 送信機
- 1 1 6 3 タグ挿入器
- 1 1 6 4 メインコントローラ
- 1 1 6 5 安定タイマ
- 1 1 6 6 到着間隔タイマ
- 1 1 6 7 コスト参照タイマ
- 1 1 6 8 関数演算器
- 1 1 6 9 平滑化回路
- 1 1 5 1 1 タグ削除器
- 1 1 5 1 2 B P D U 送信機
- 1 1 5 1 3 タグ挿入器
- 1 1 5 1 4 ツリーコントローラ
- 1 1 5 1 4  $\alpha$  ツリーコントローラ
- 1 1 5 1 4  $\beta$  ツリーコントローラ
- 1 1 5 1 4  $\gamma$  ツリーコントローラ
- 1 1 5 1 5 ツリーテーブル

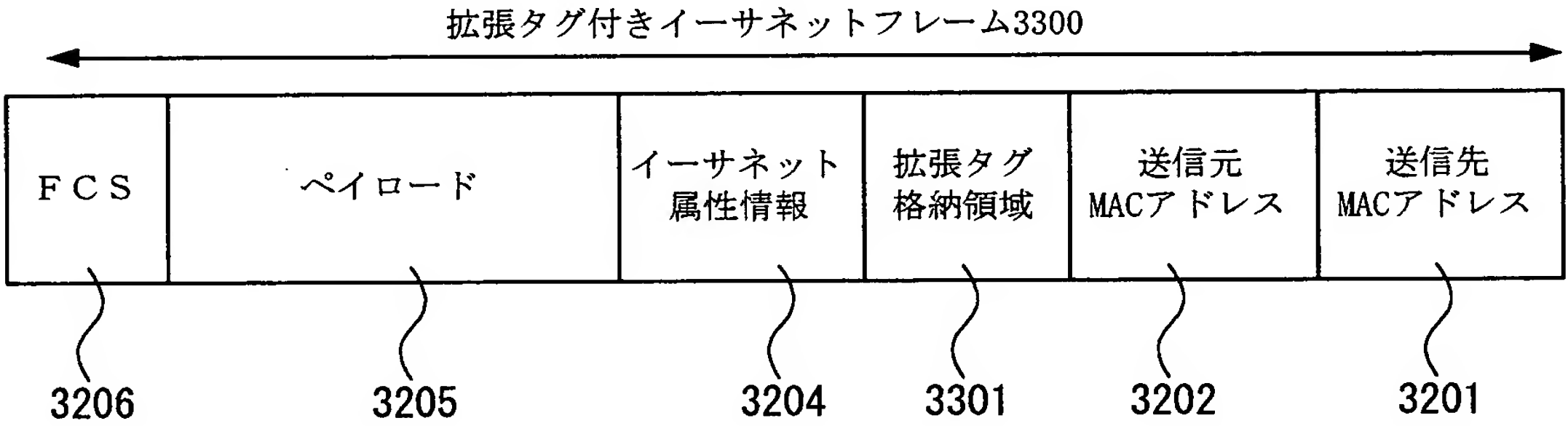
1 1 5 1 6 コスト操作器  
1 1 6 4 1 ~ 1 1 6 4 9 状態  
1 1 6 4 A 状態  
1 1 6 4 B 状態  
1 2 0 障害検出器  
2 8 0 0 T P I D  
2 8 0 1 T C I  
2 8 0 2 P r i o r i t y (プライオリティ)  
2 8 0 3 C F I  
2 8 0 4 V L A N I D  
3 5 0 8 ブロードキャストタグ (拡張タグ)  
5 0 0 1 拡張タグ識別領域  
5 0 0 2 拡張タグ情報領域  
5 0 0 3 P r i o r i t y、タグタイプフィールド  
5 0 0 4 拡張タグ情報フィールド  
6 1 0 1 ノード  
6 1 0 2 ルートポート  
6 1 0 3 オルタネートポート  
6 1 0 4 デイジグネイテッドポート

【書類名】 図面

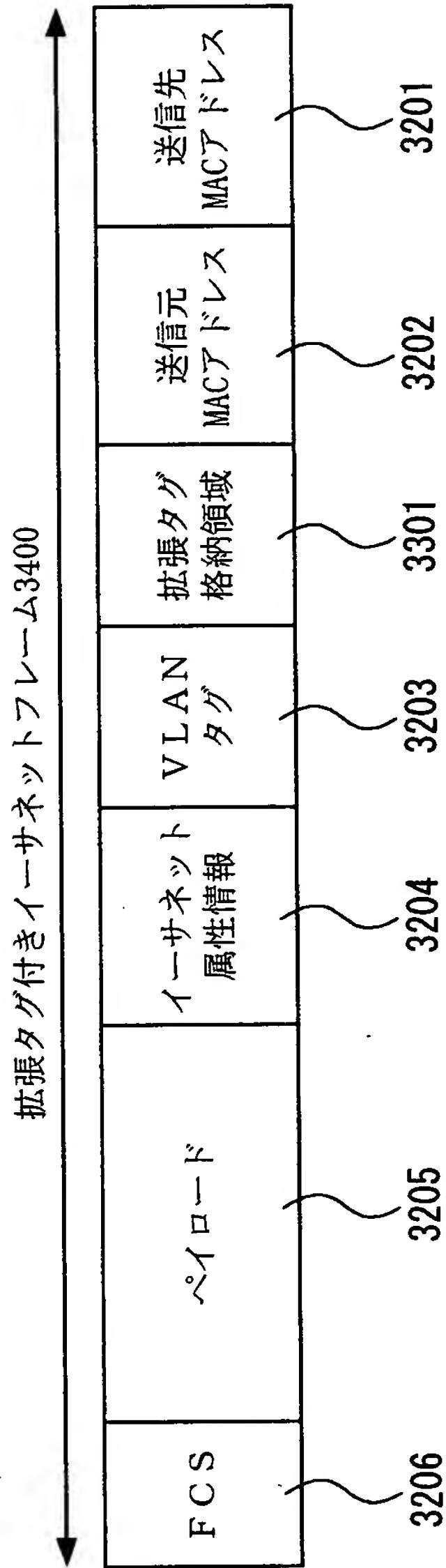
【図 1】



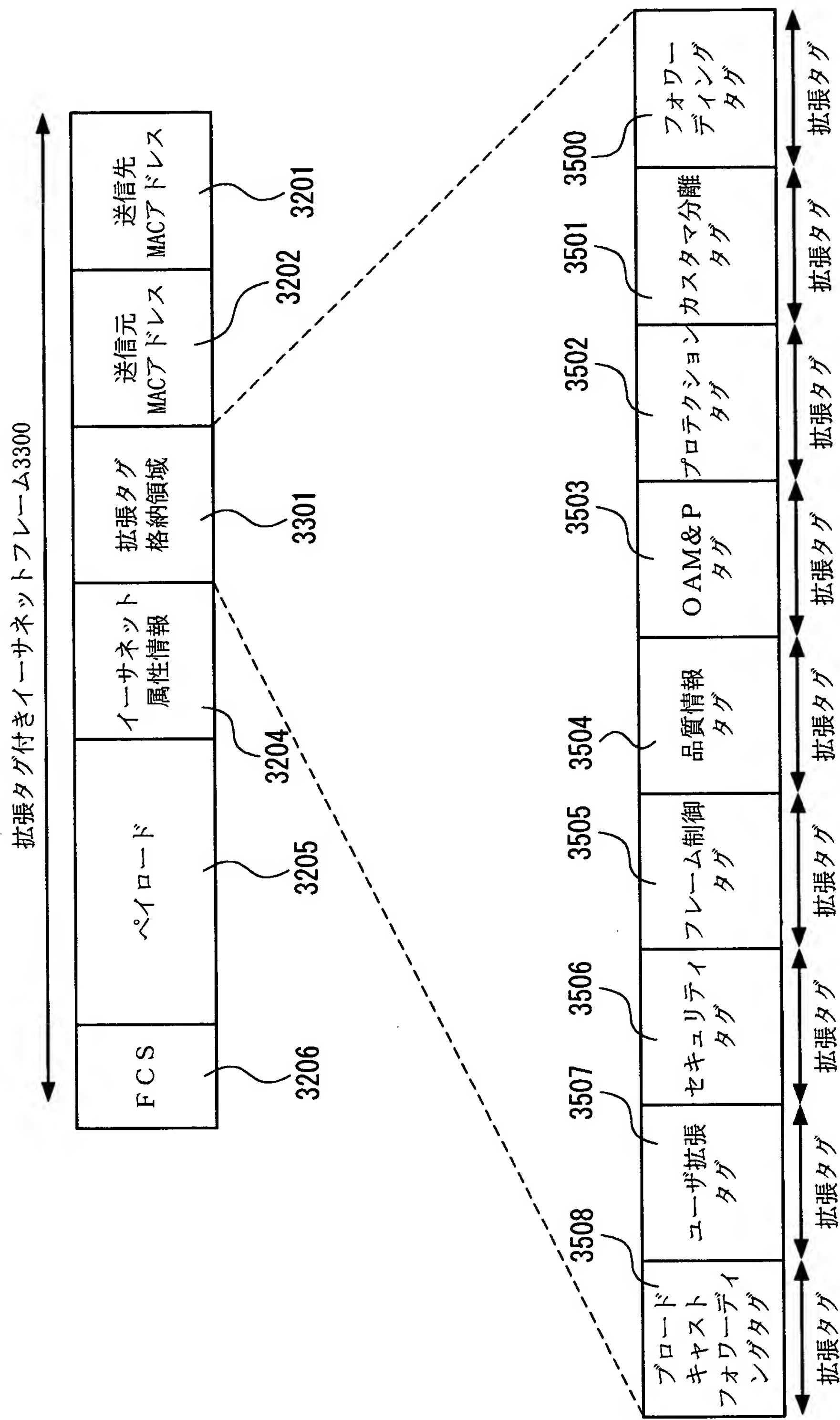
【図 2】



【図 3】



【図 4】

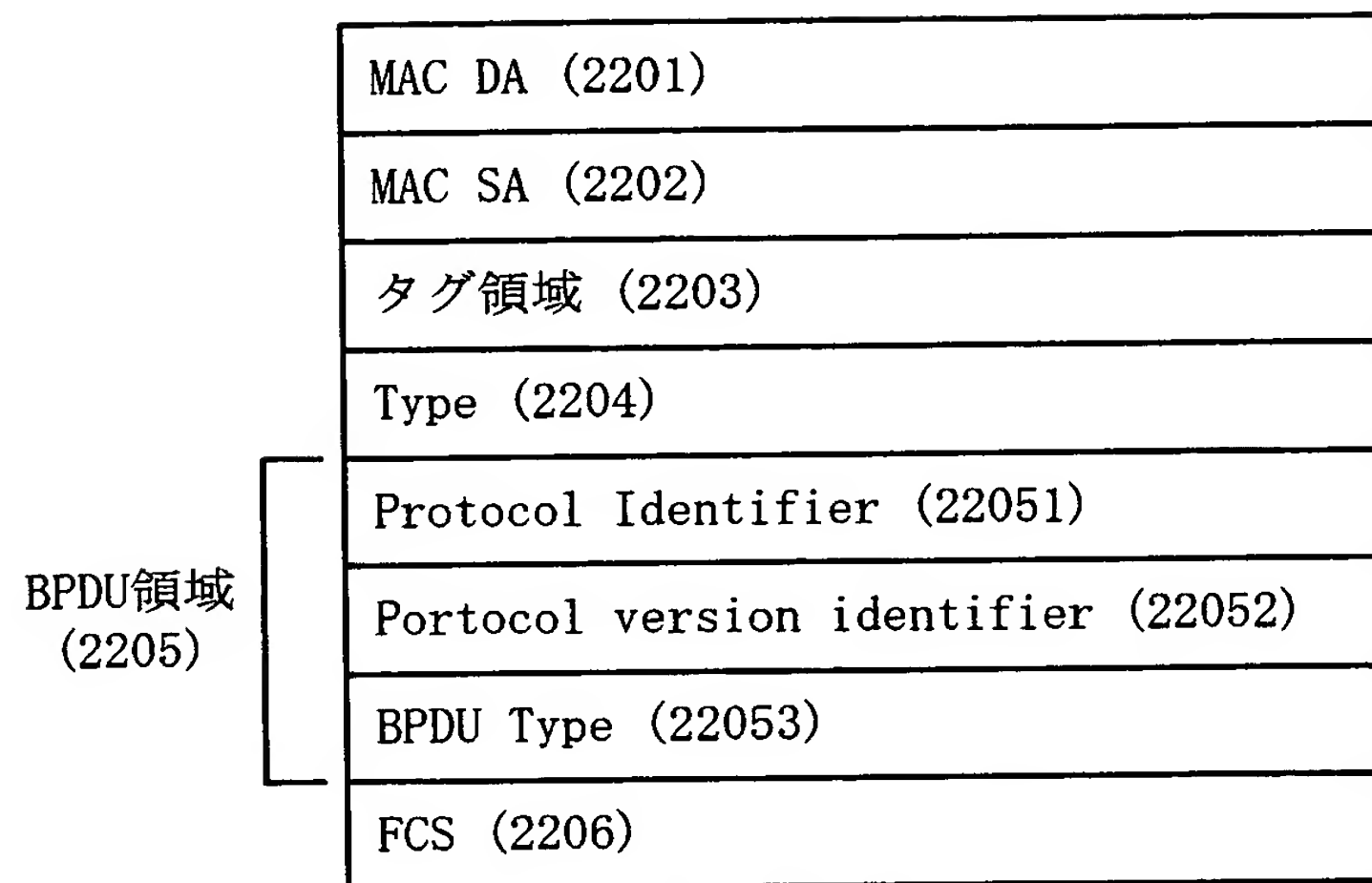


【図 5】

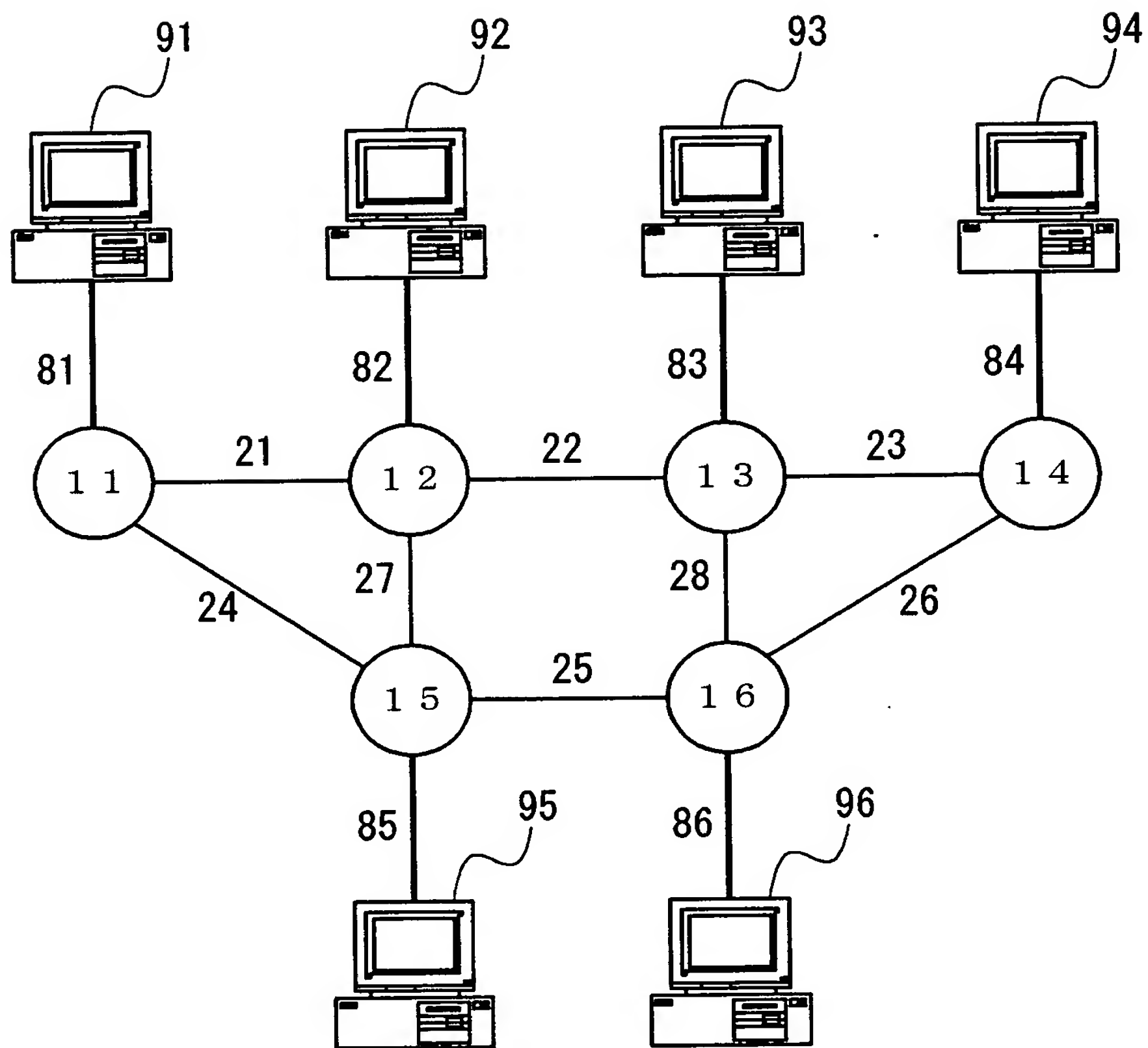
BPDU領域 (2205)	MAC DA (2201)
	MAC SA (2202)
	タグ領域 (2203)
	Type (2204)
	Protocol Identifier (22051)
	Portocol Version identifier (22052)
	BPDU Type (22053)
	Flags (22054)
	Root Identifier (22055)
	Root Path Cost (22056)
	Bridge Identifier (22057)
	Port Identifier (22058)
	Message Age (22059)
	MAX Age (2205A)
	Hello Time (2205B)
	Forward Deley (2205C)
	FCS (2206)



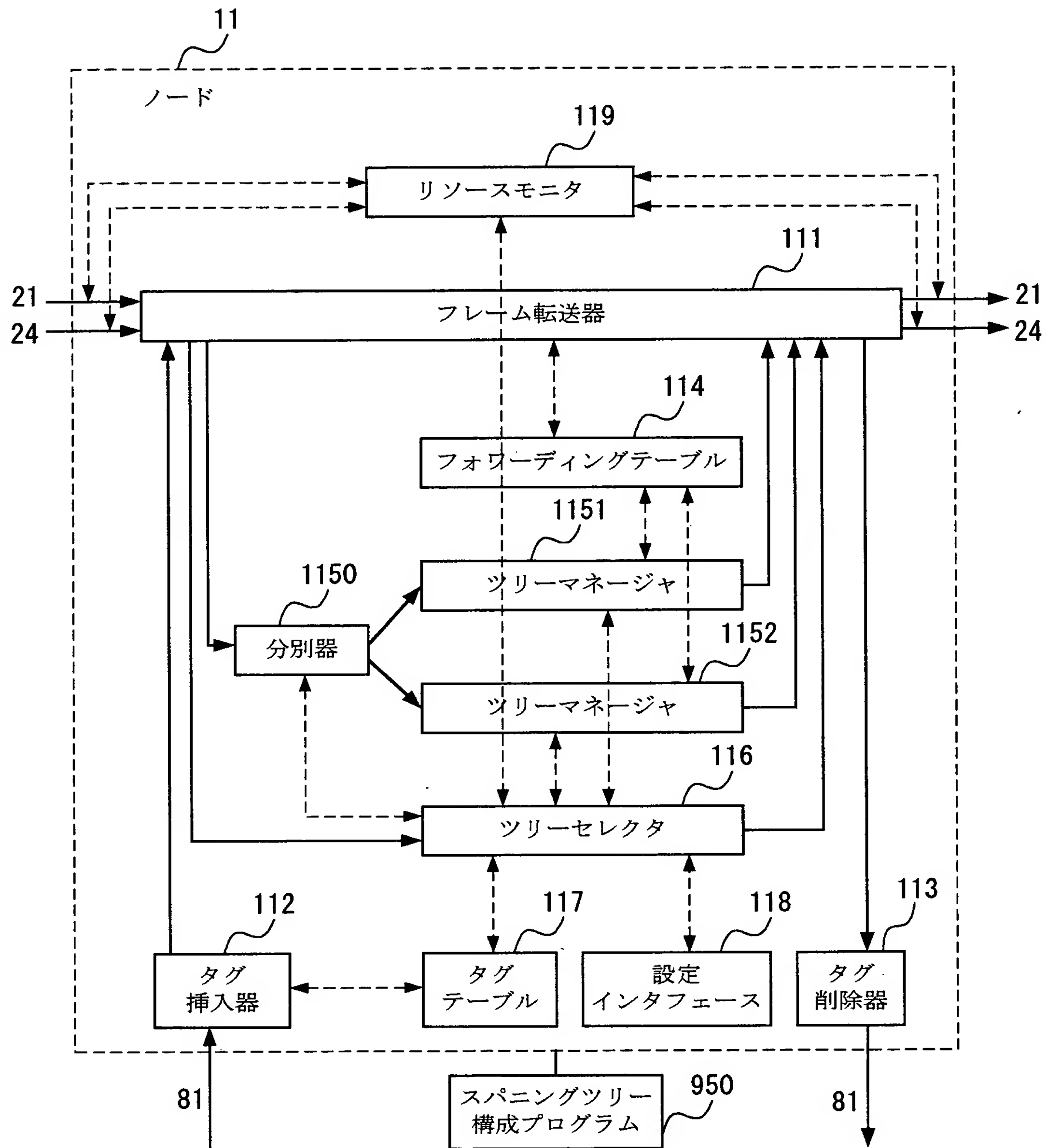
【図 6】



【図 7】



【図 8】

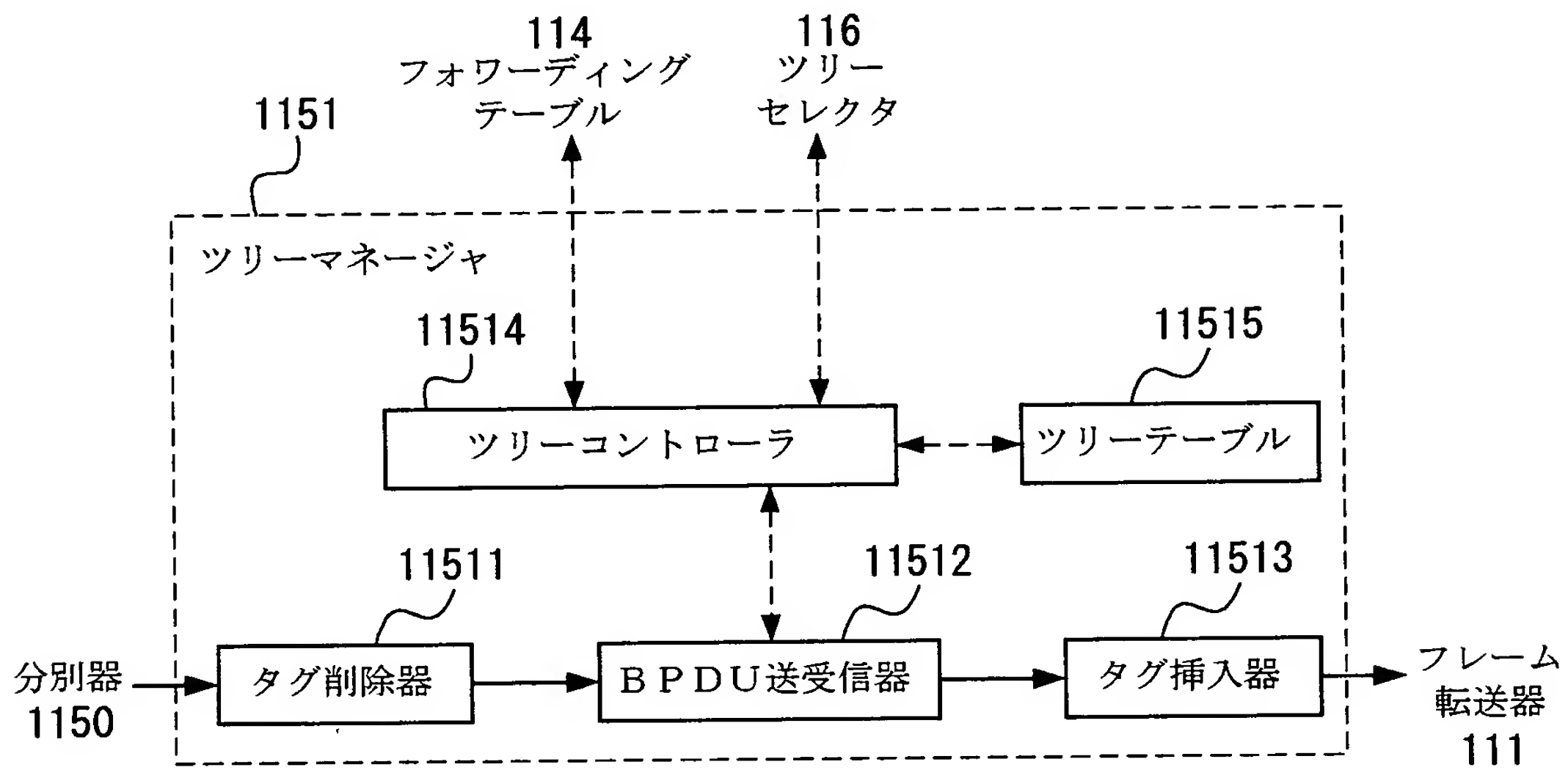


【図 9】

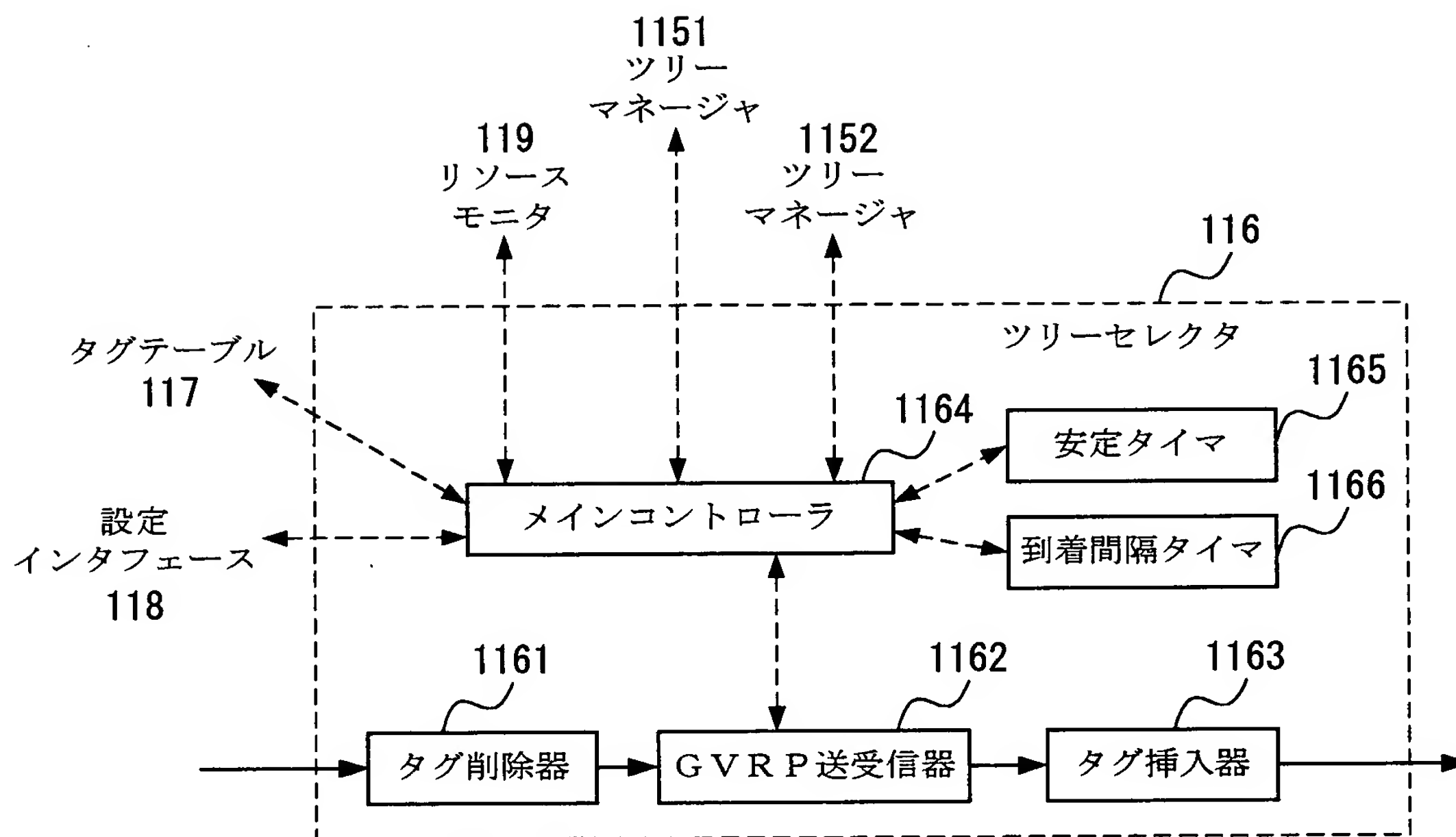
114 フォワーディングテーブル

タグ 1 1 4 1	出力ポート 1 1 4 2
現用系タグ	P o r t 2 (例)
予備系タグ	P o r t 1 (例)

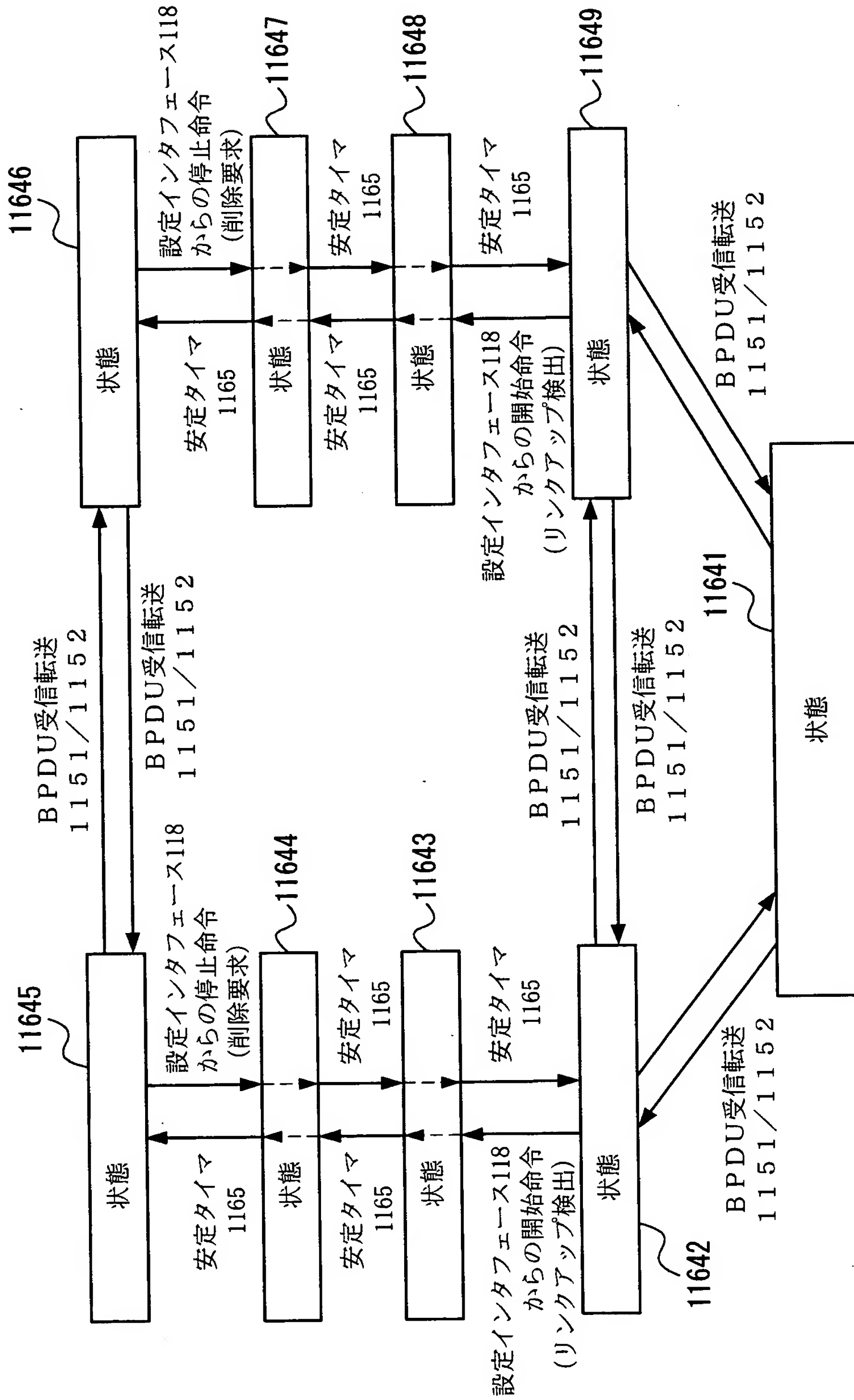
【図 1 0】



【図 1 1】



【図 1 2】



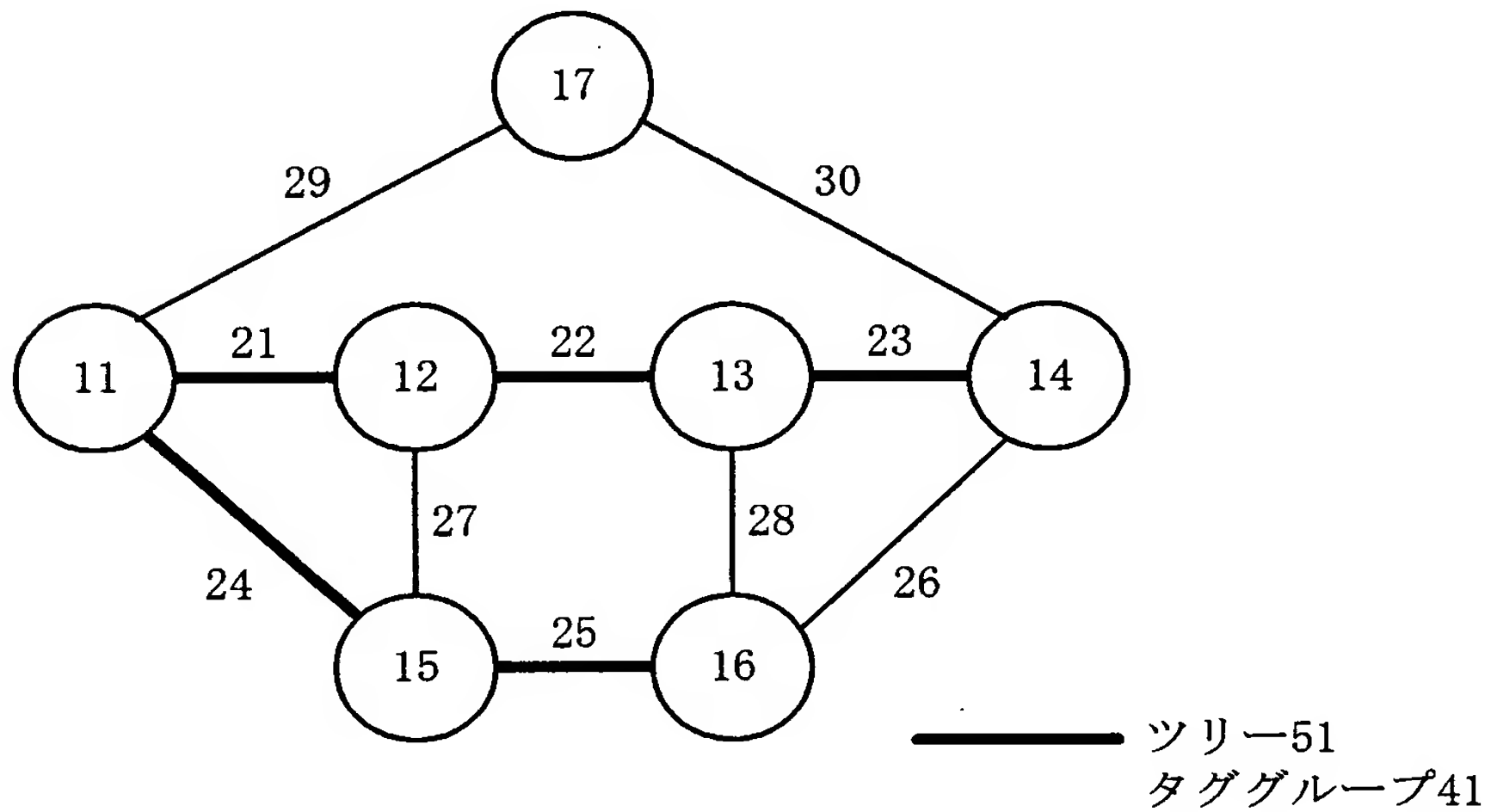


【図 1 3】

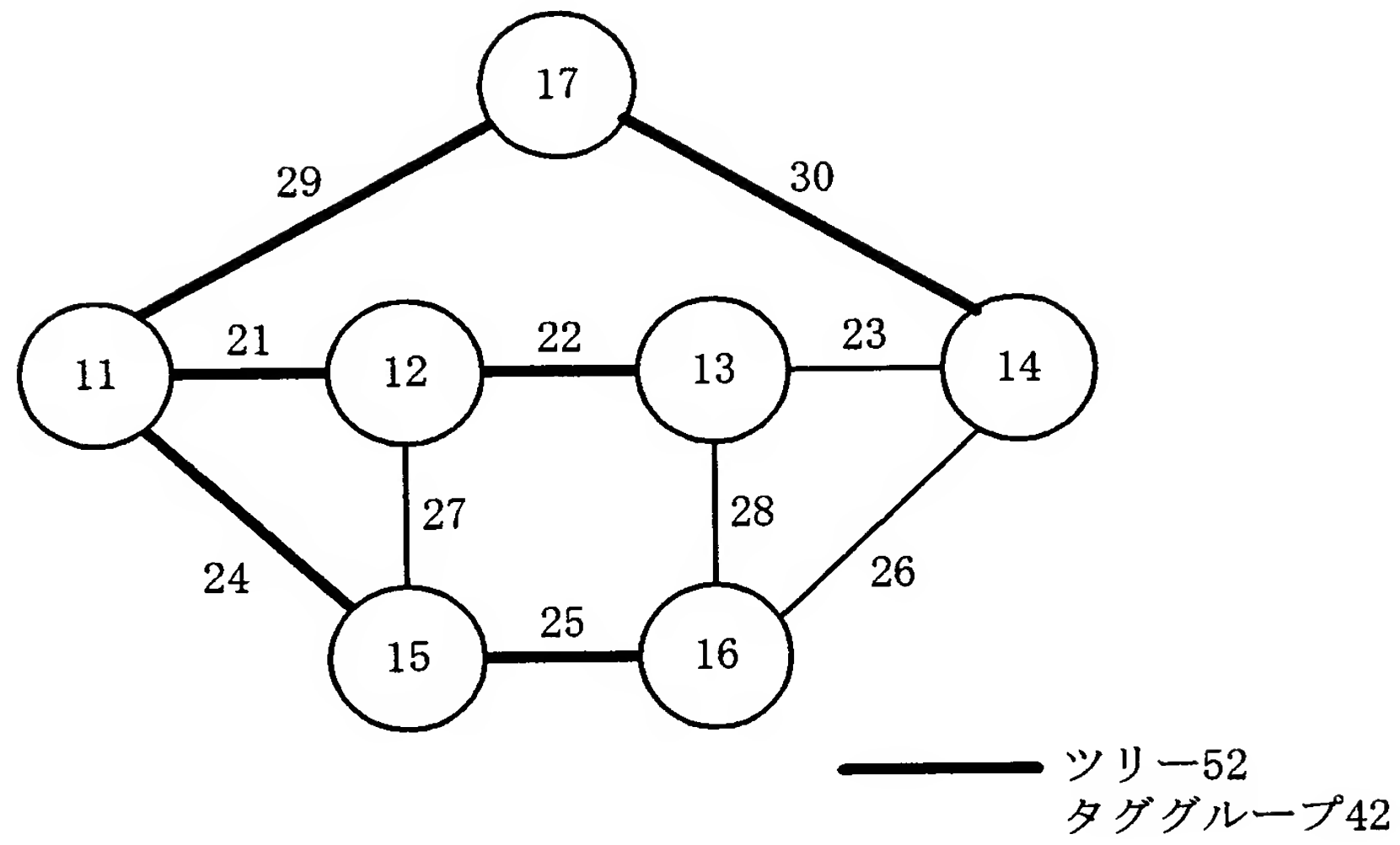
117 タグテーブル

宛先MACアドレス1171	挿入タグ1172
ネットワーク全体の全てのノードを表す値	現在、現用系となっているタグを表すタググループのID

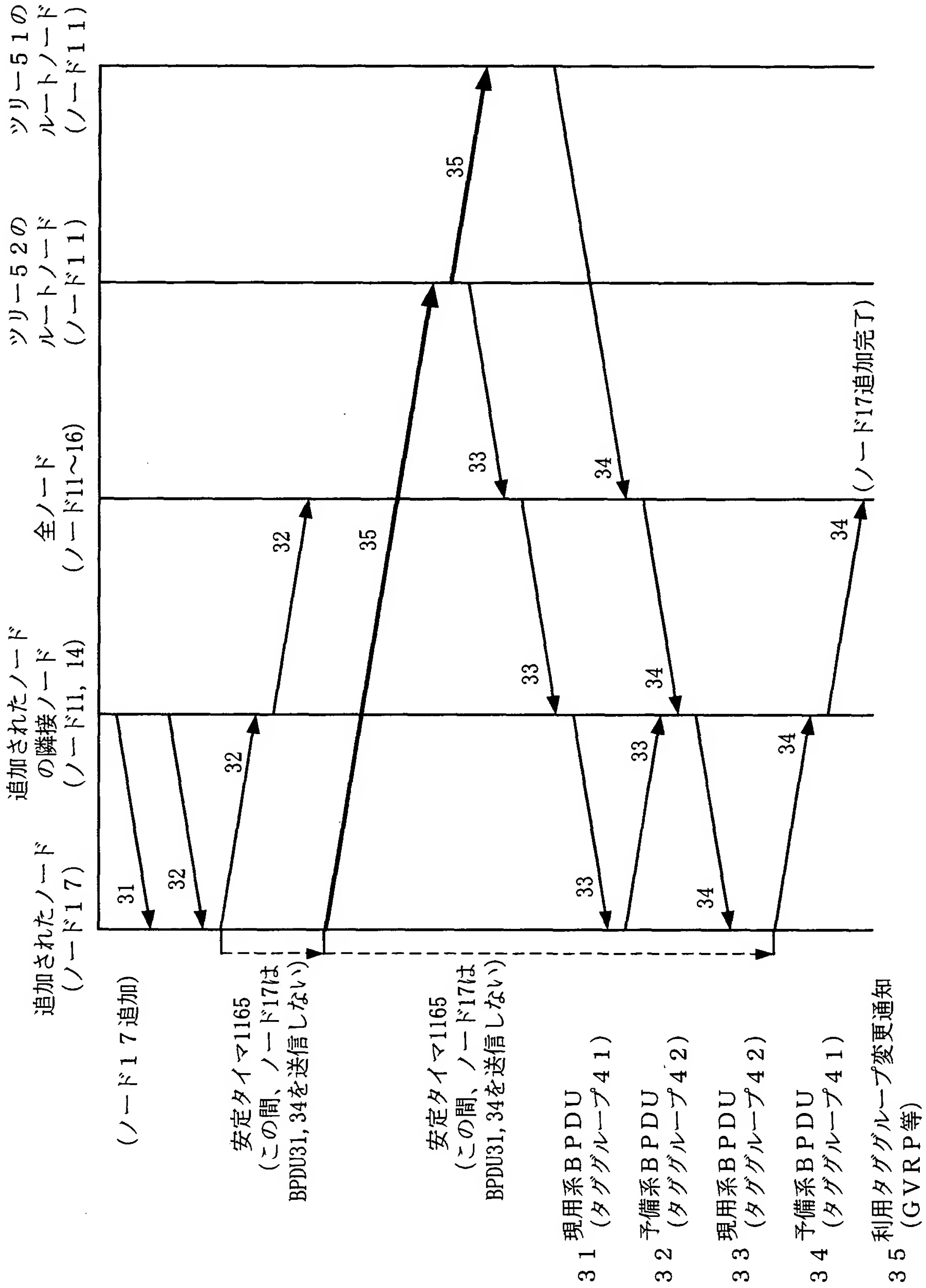
【図 1 4】



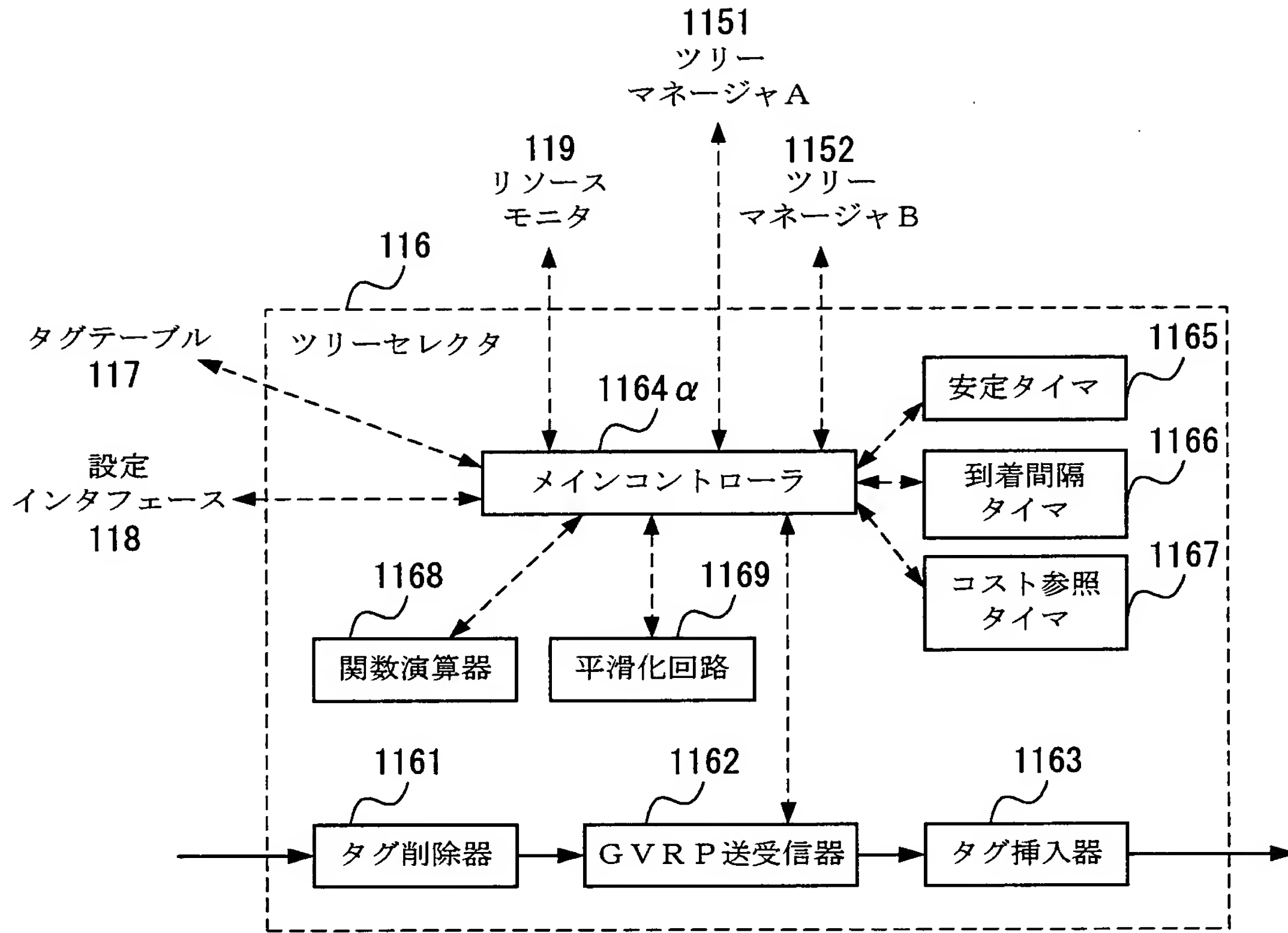
【図 1 5】



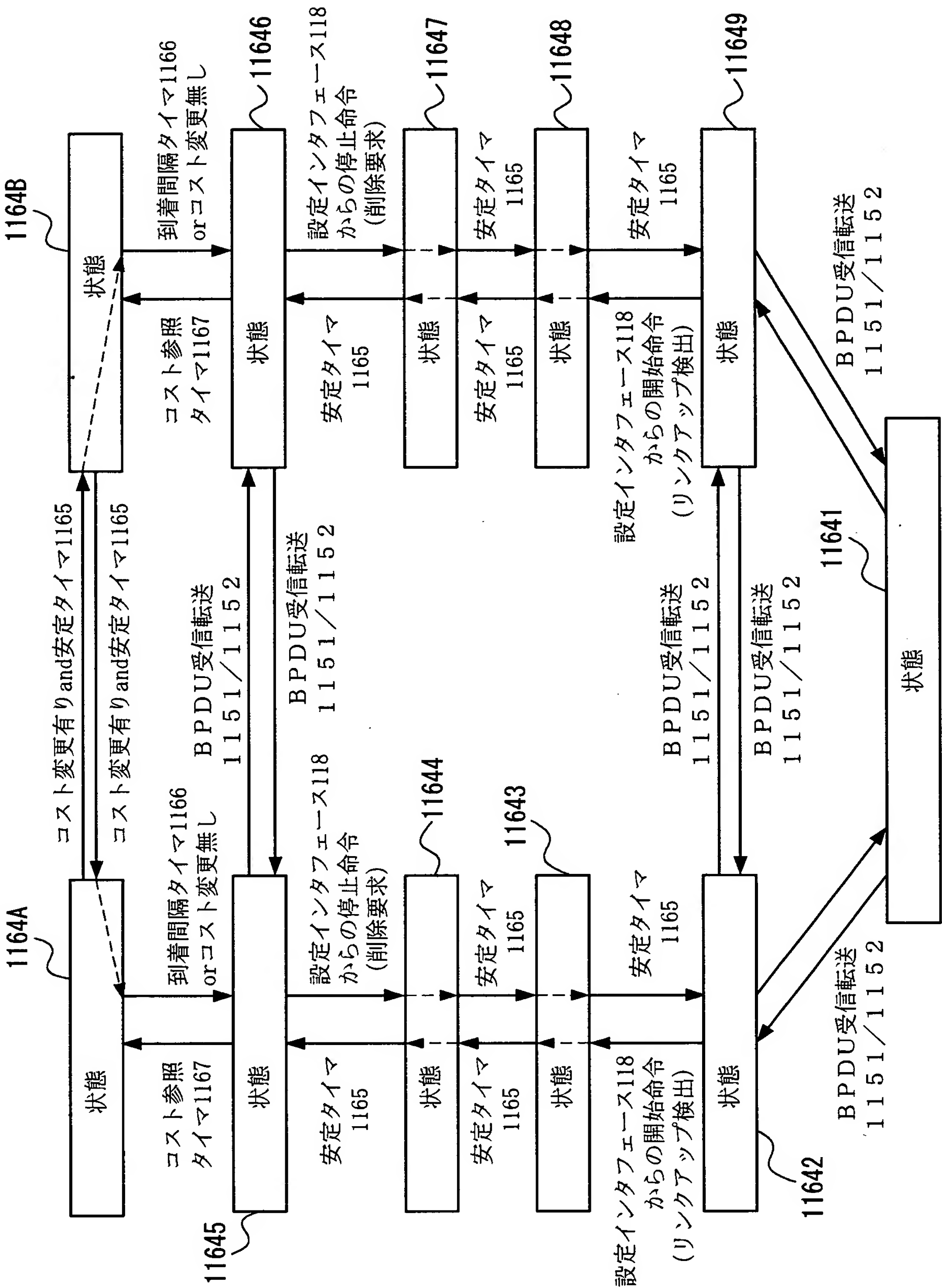
【図 1 6】



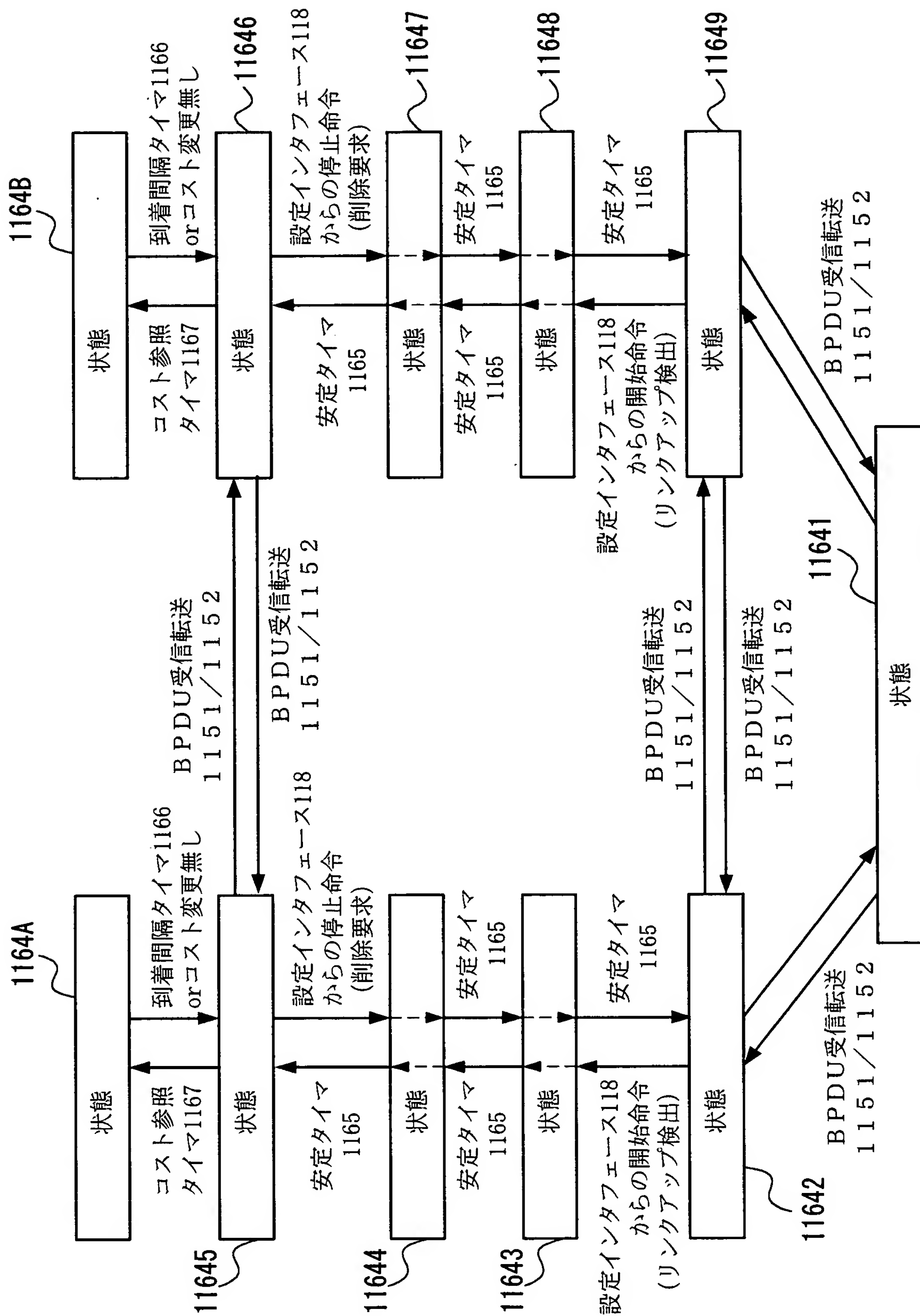
【図 1 7】



【図 1 8】

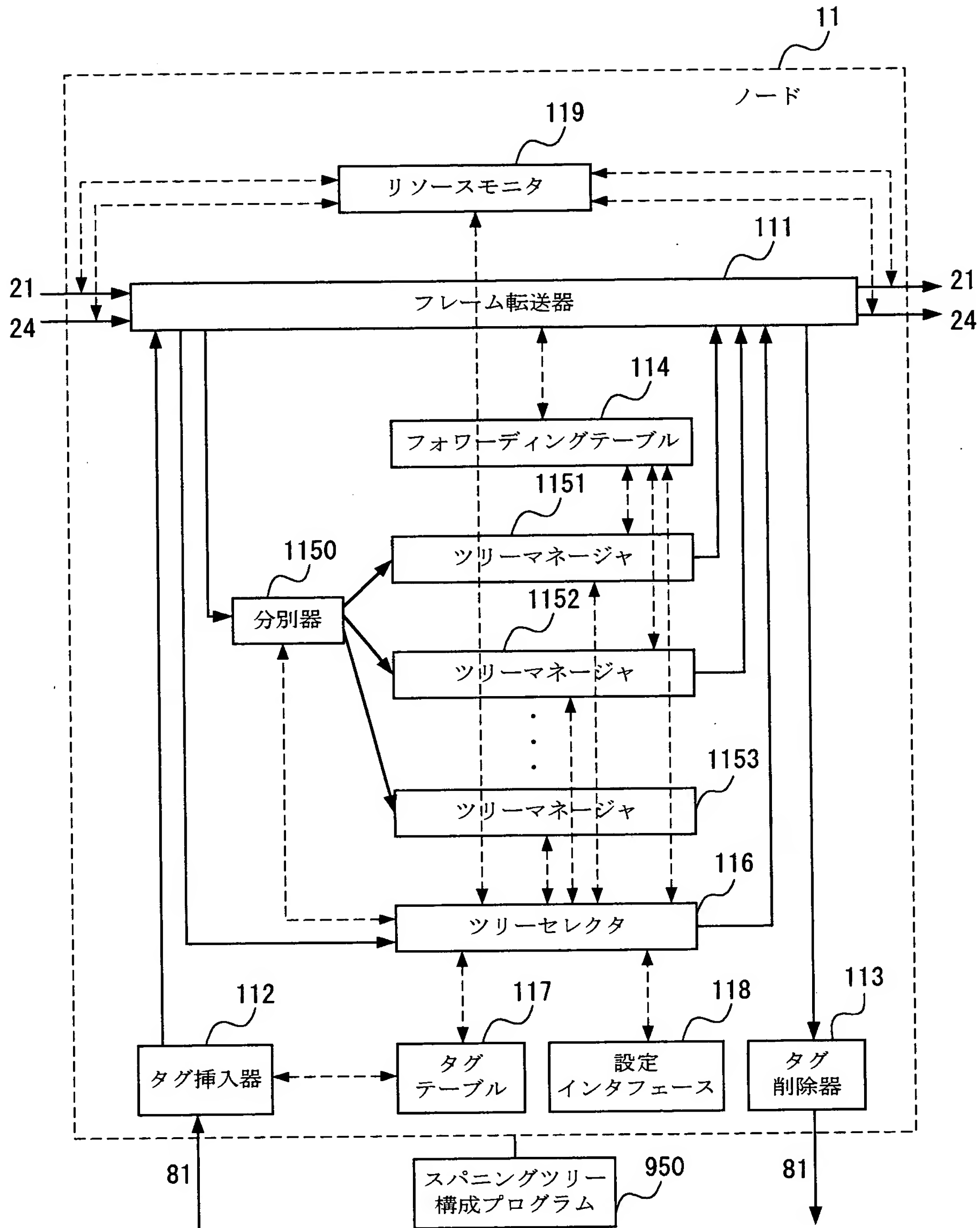


【図 1 9】





【図 2 0】



【図 2 1】

114 フォワーディングテーブル

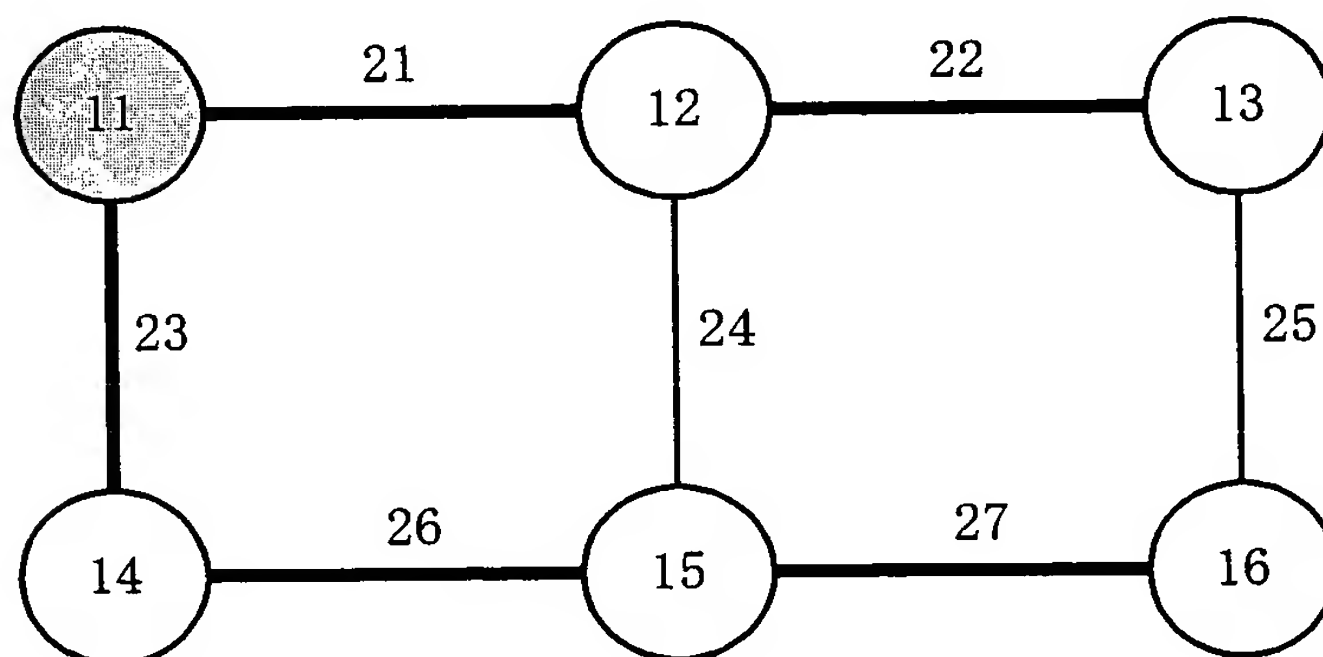
タグ 1 1 4 1	出力ポート 1 1 4 2
4 1	P o r t 1
4 2	P o r t 2
4 3	P o r t 2
4 4	P o r t 1

【図 2 2】

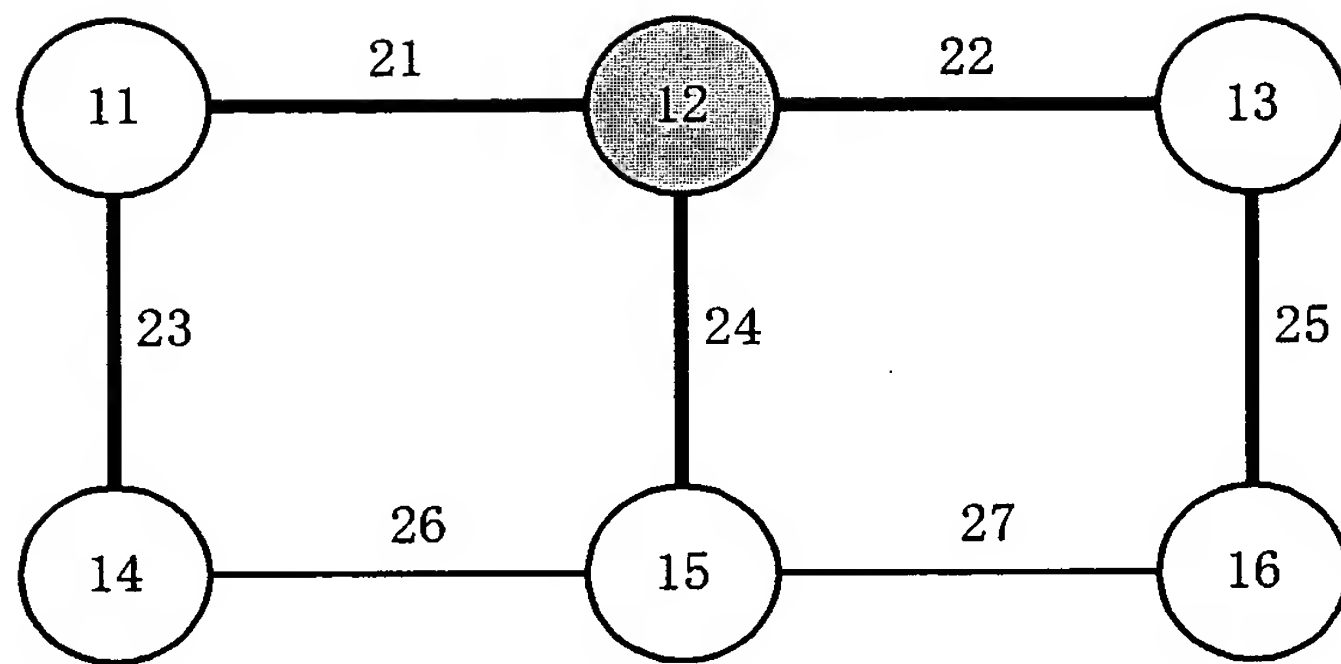
117 タグテーブル

宛先MACアドレス 1 1 7 1	挿入タグ 1 1 7 2
9 1 A	4 1
9 1 C	4 1
9 2 B	4 2
9 3 X	4 3

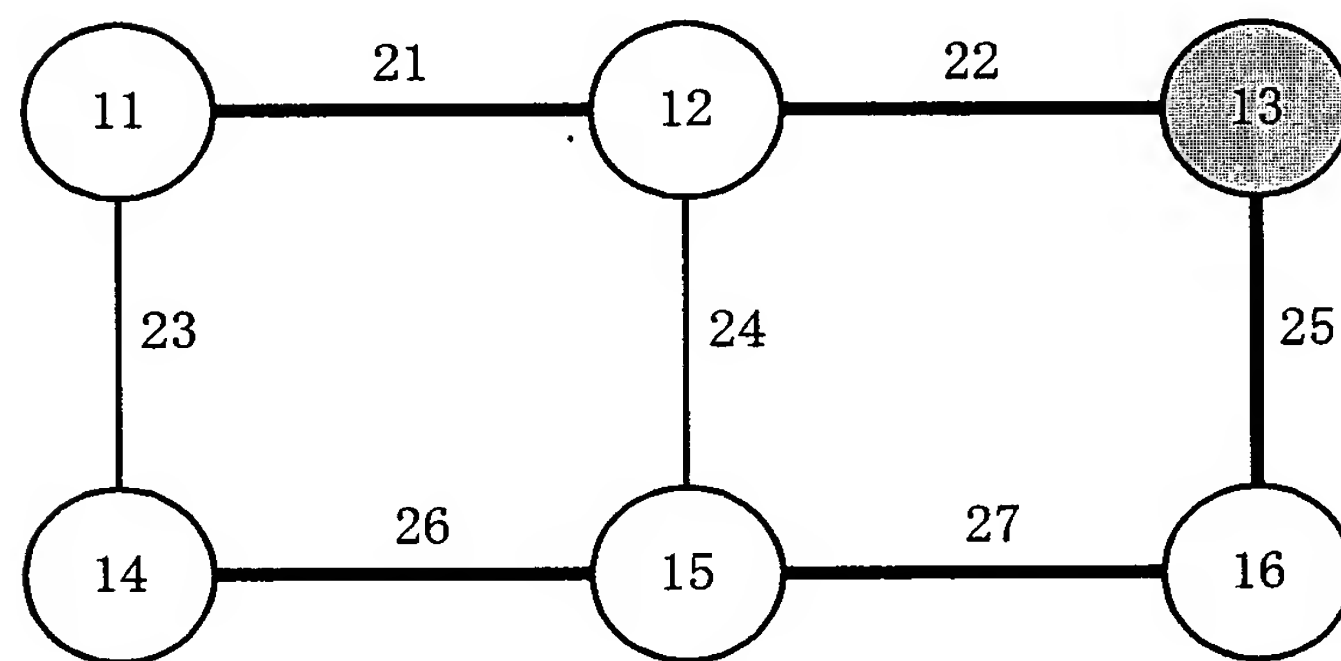
【図 2 3】



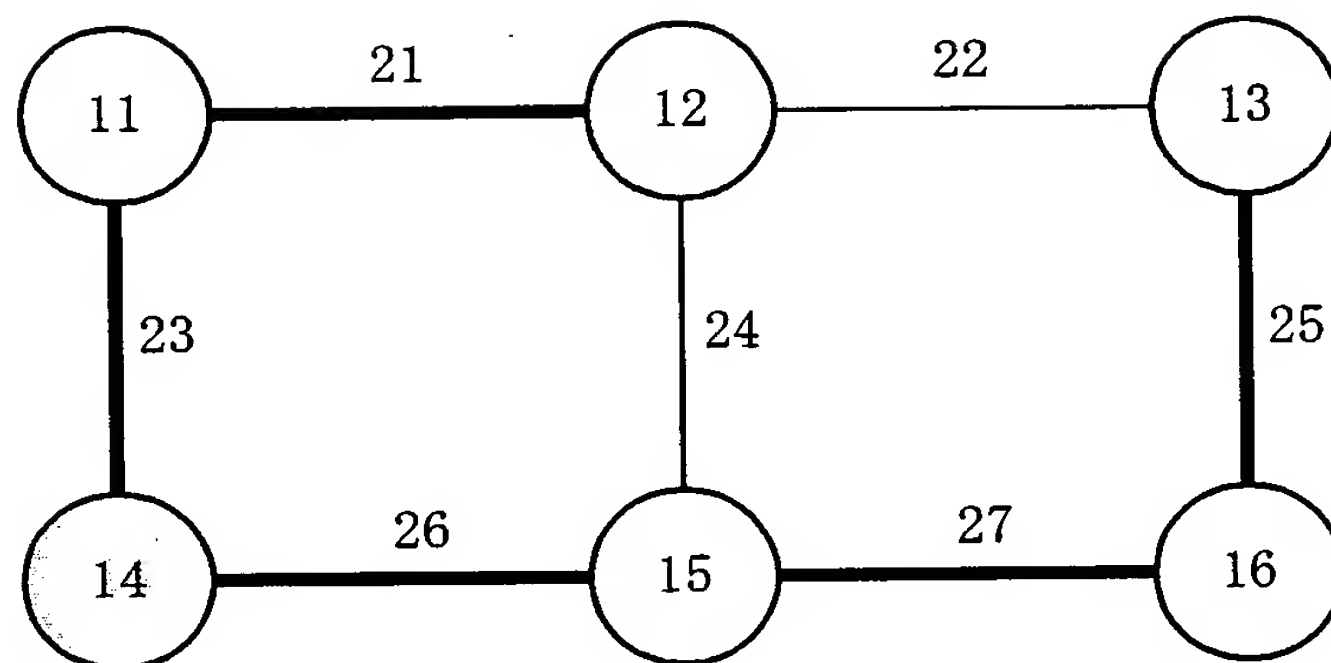
【図 2 4】



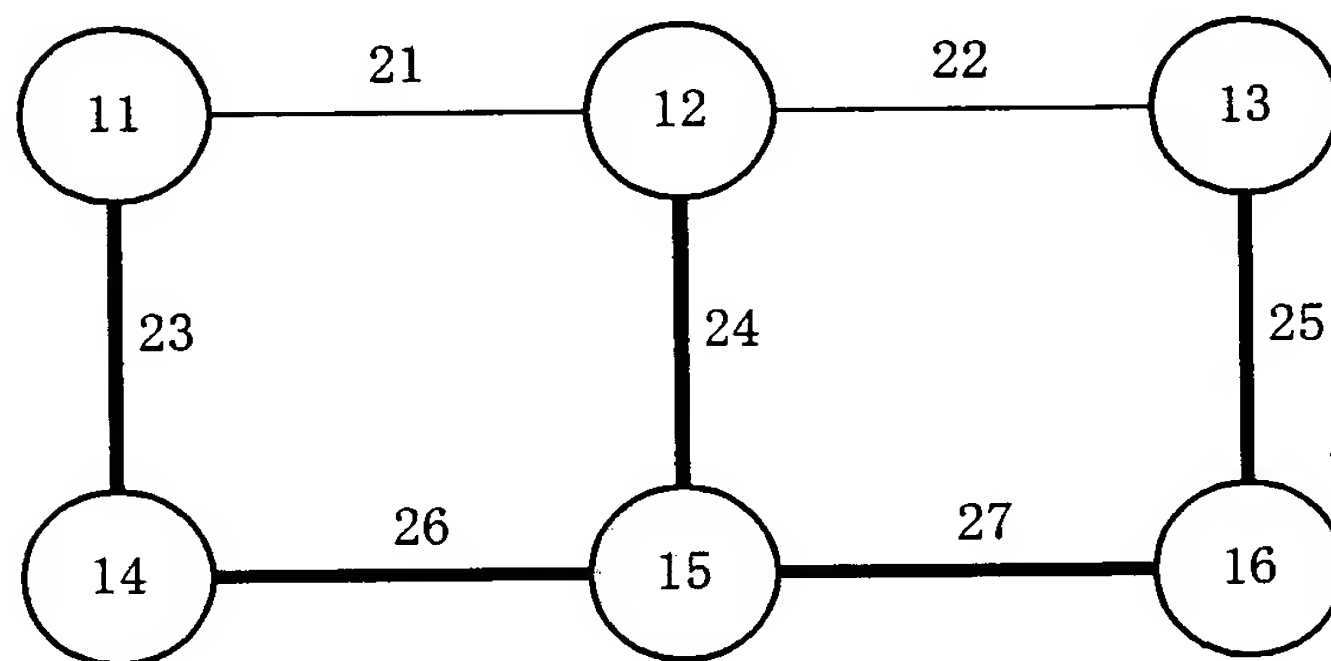
【図 2 5】



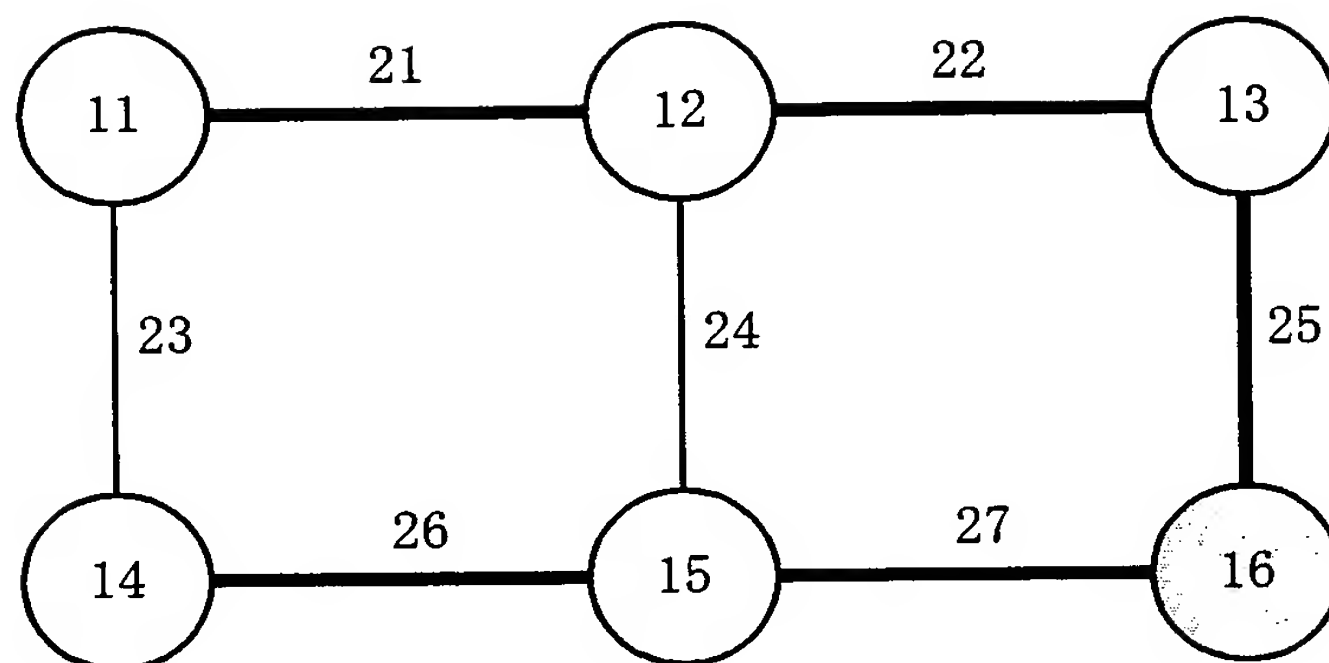
【図 2 6】



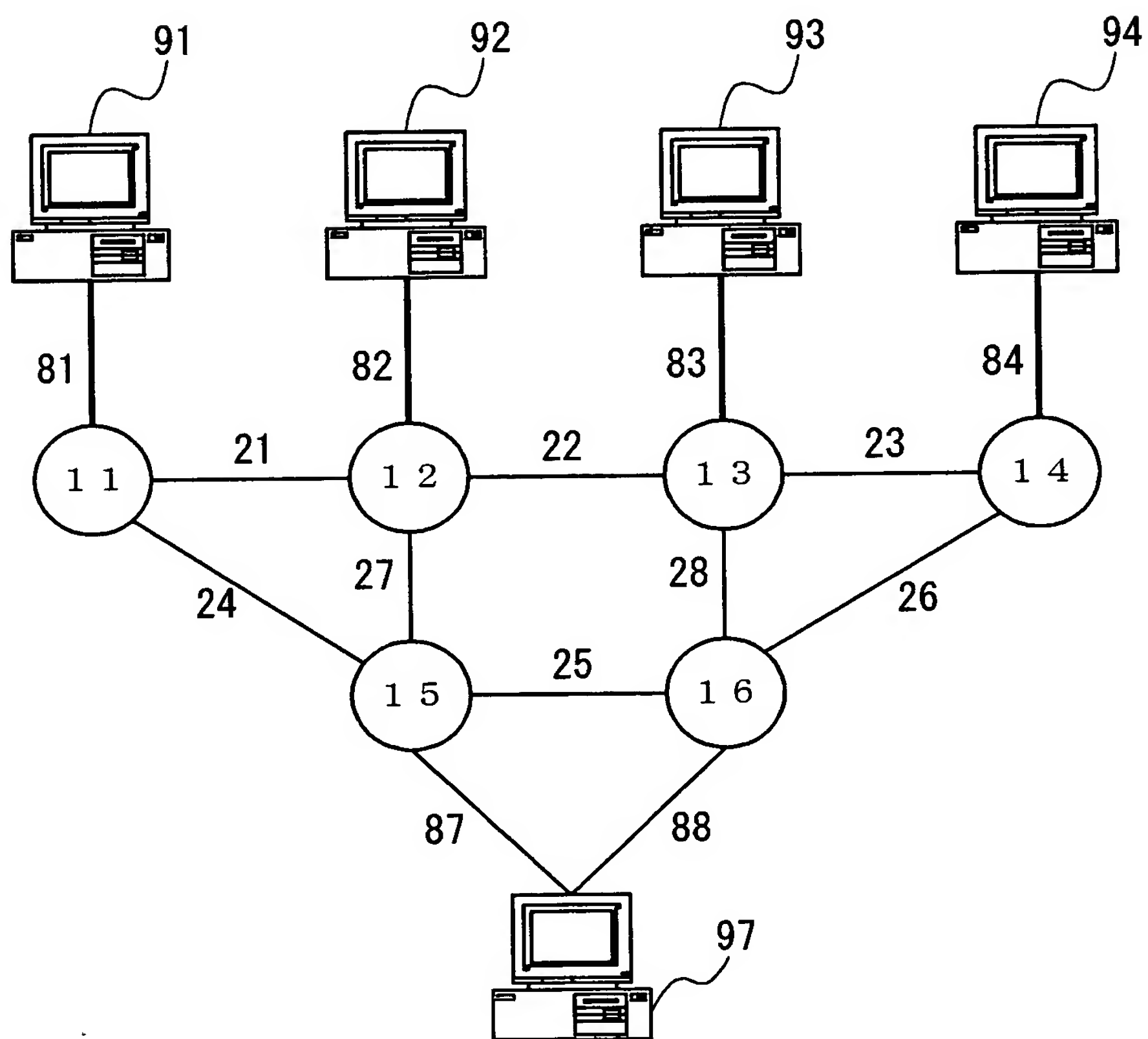
【図 2 7】



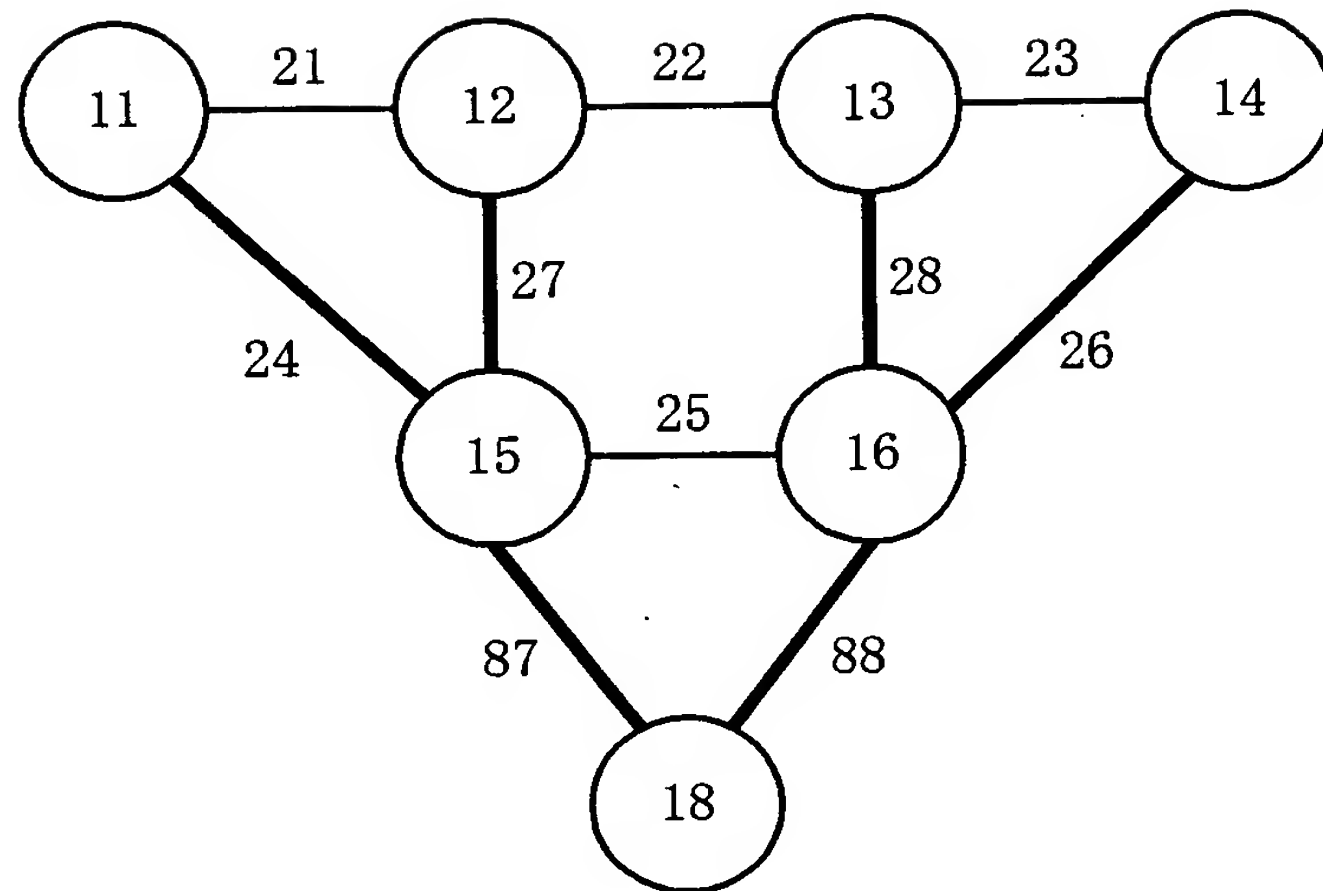
【図 2 8】



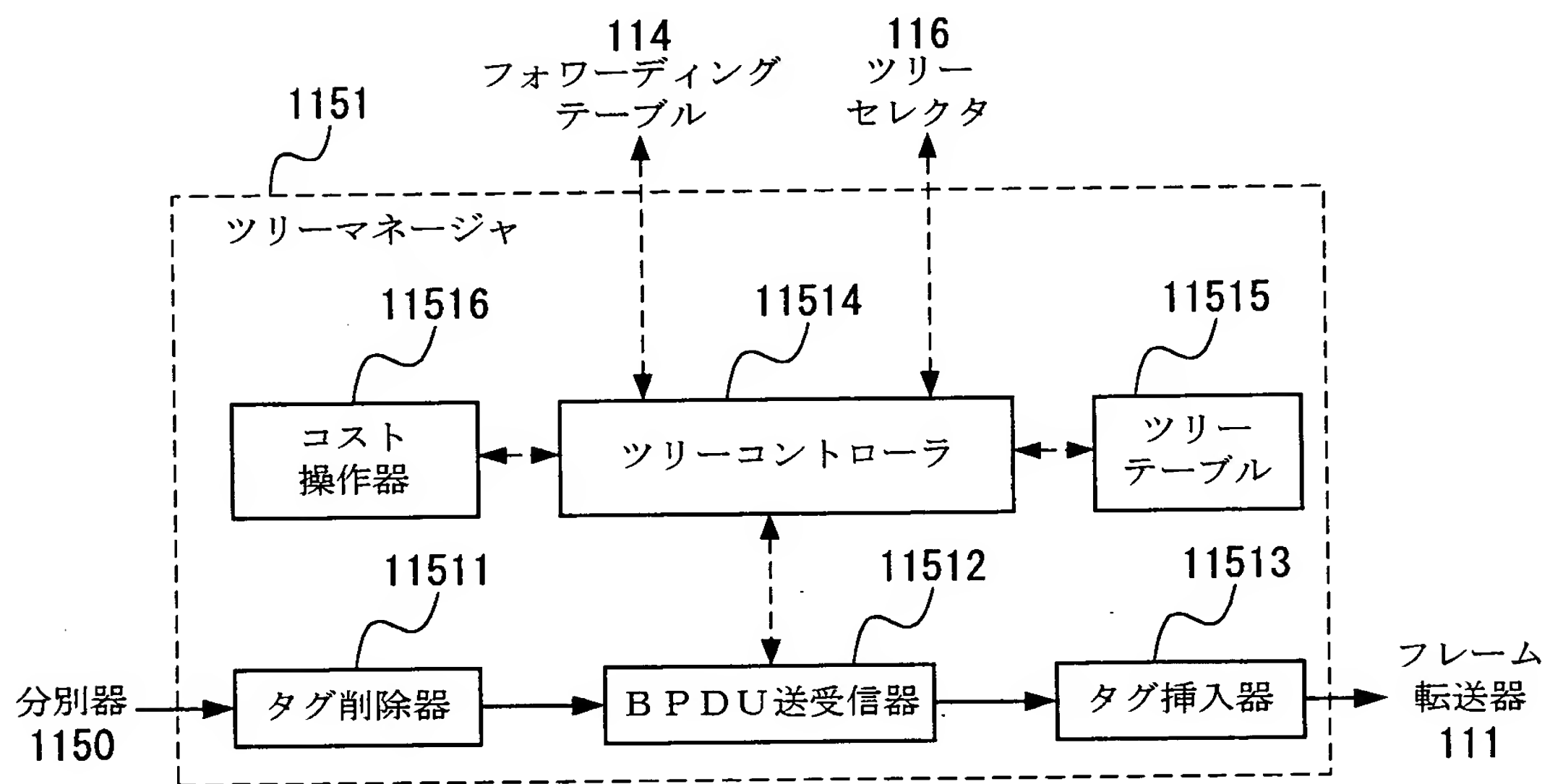
【図 2 9】



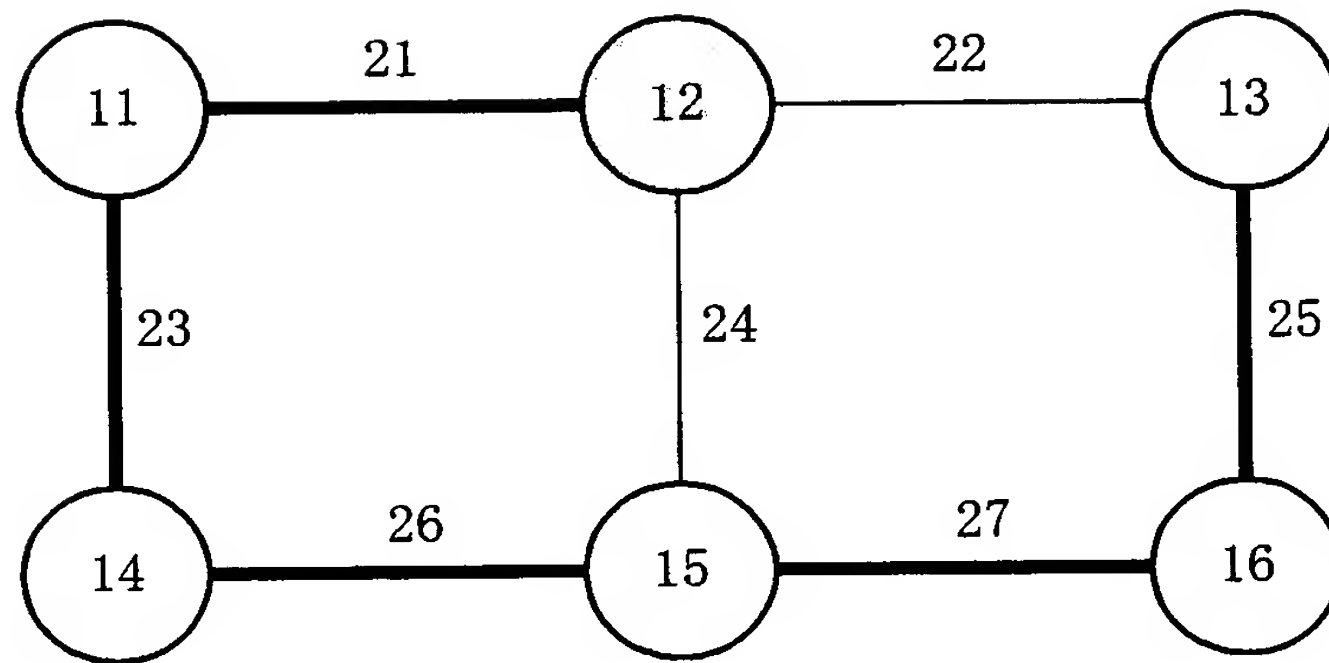
【図 3 0】



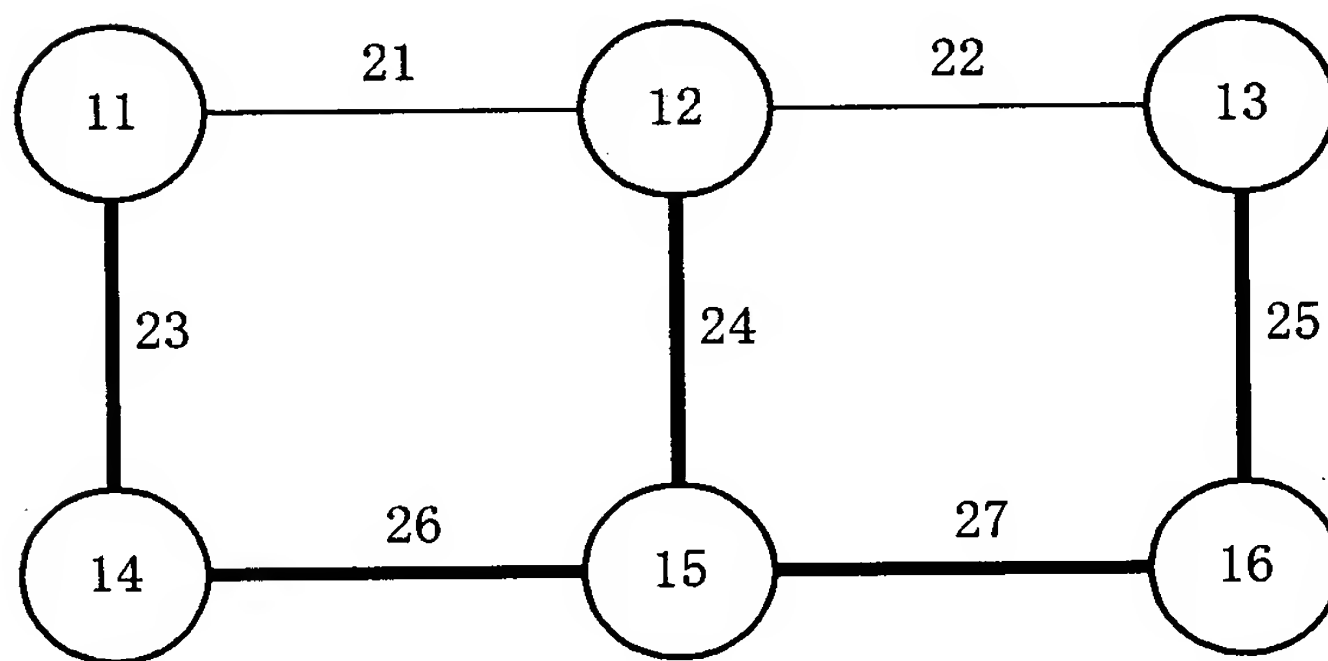
【図 3 1】



【図 3 2】

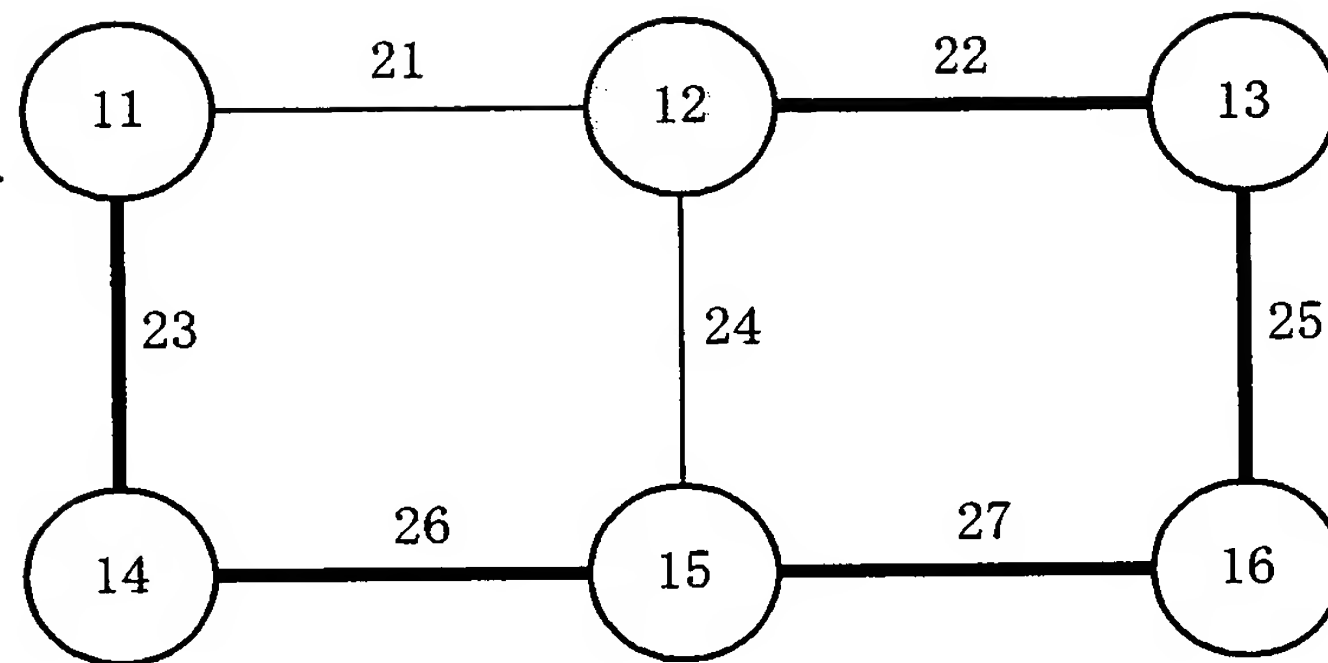


【図 3 3】

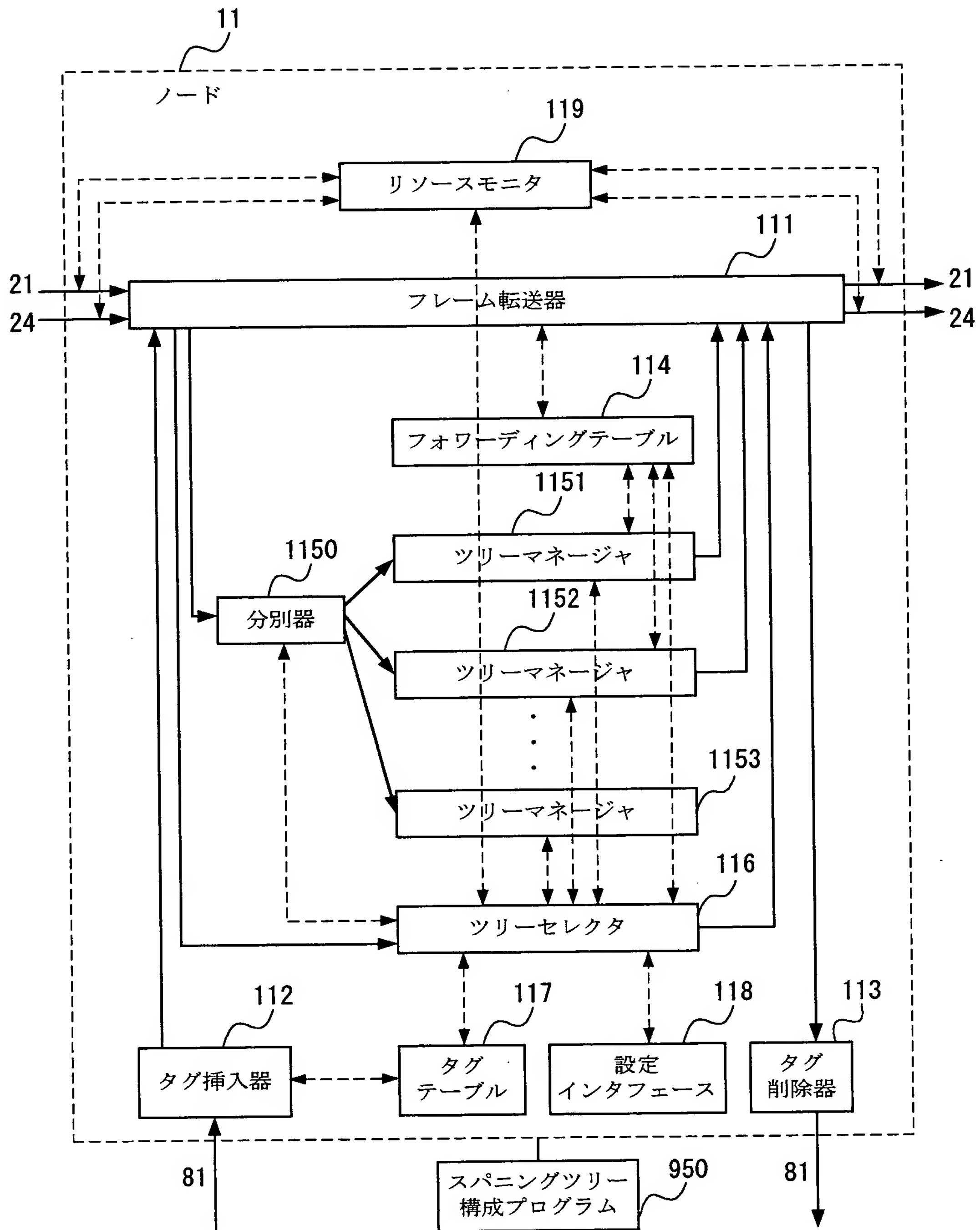




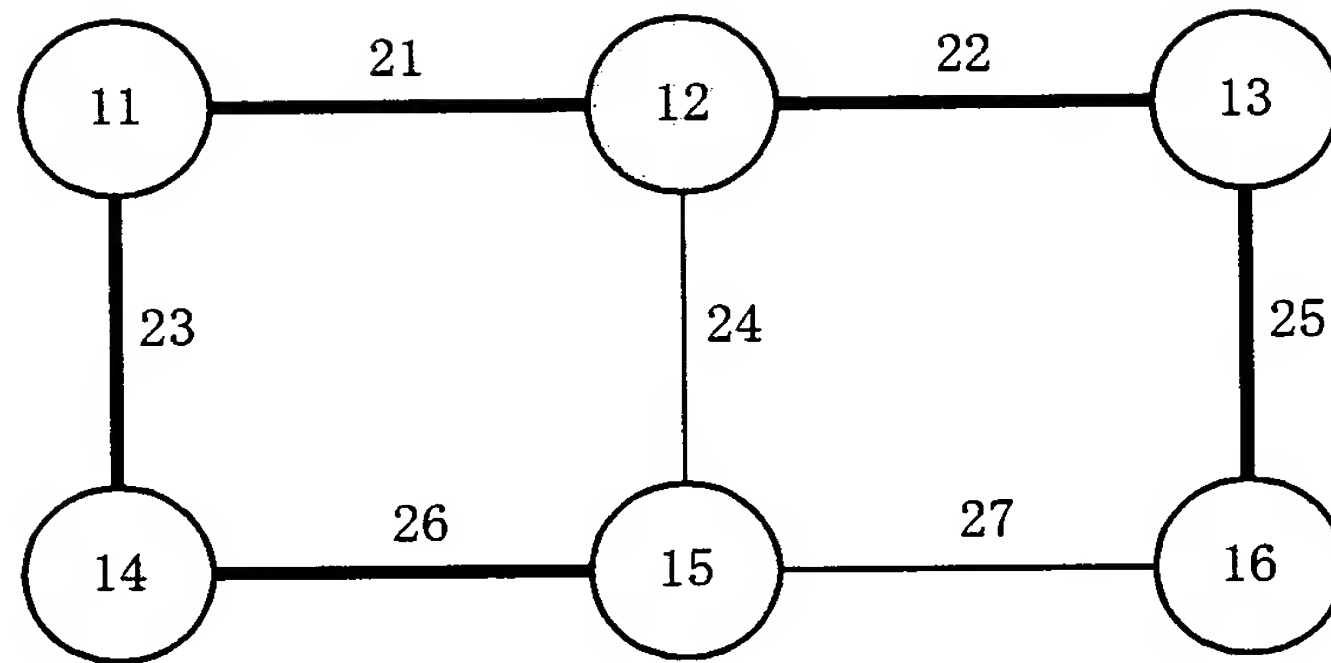
【図 3 4】



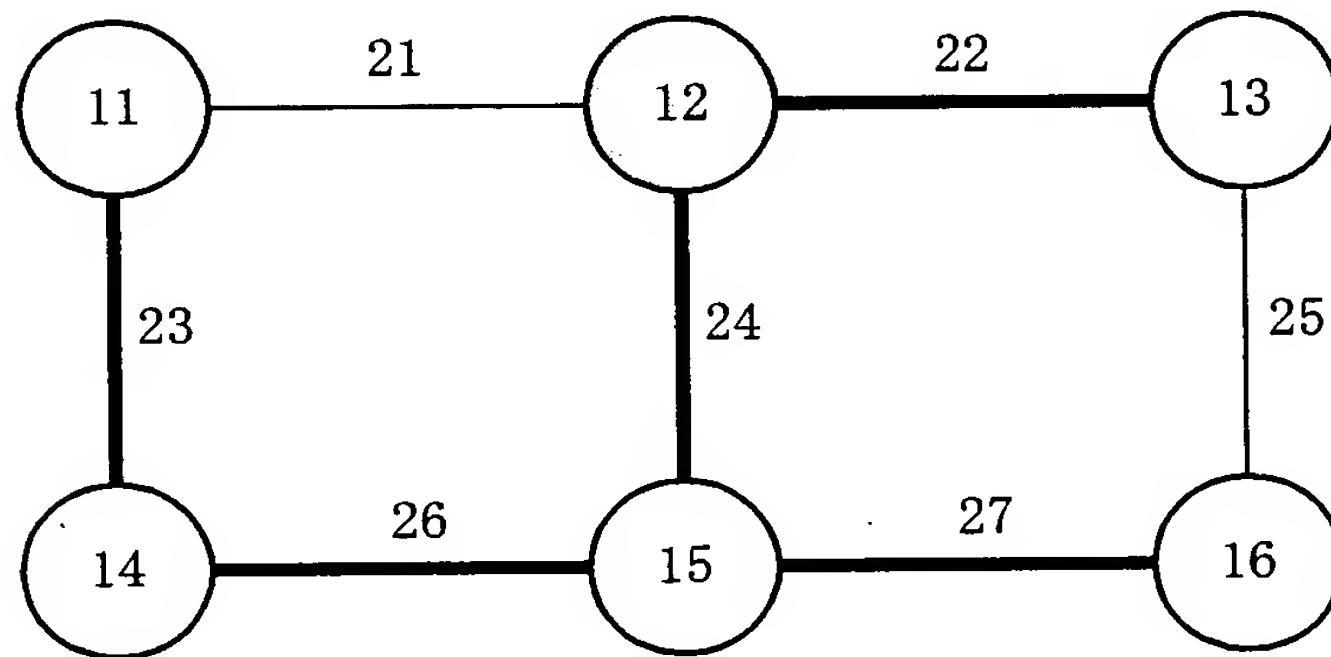
【図 3 5】



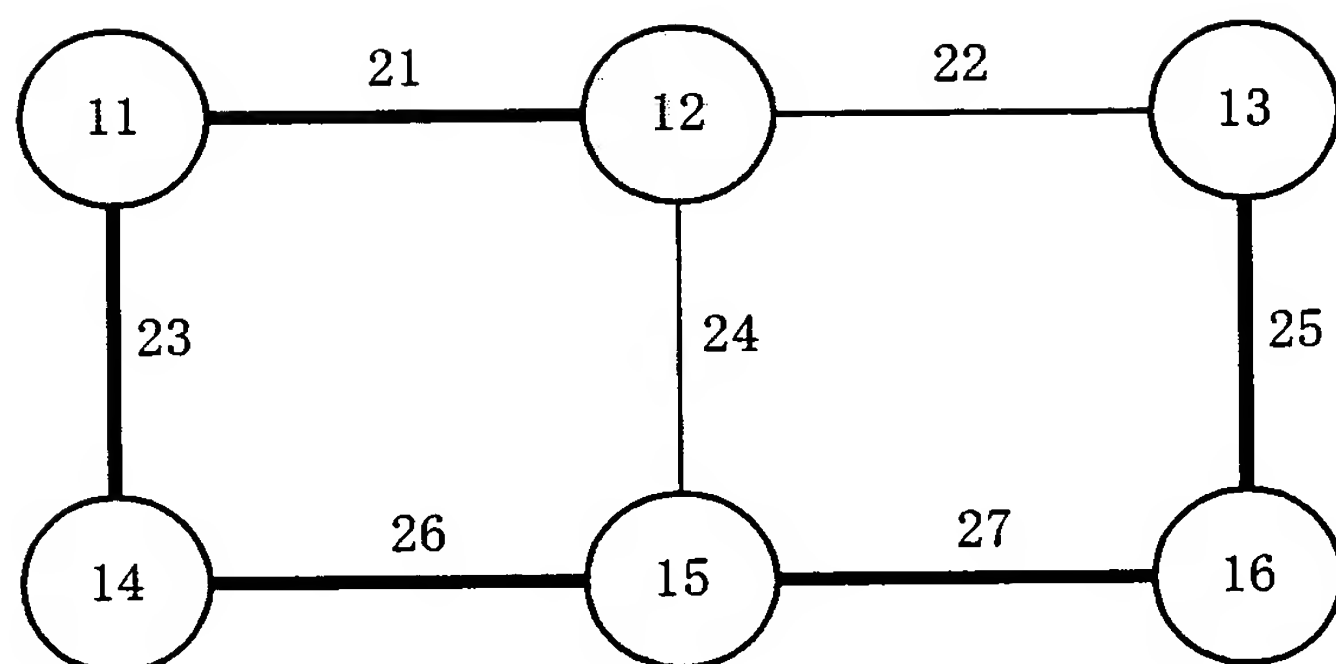
【図 3 6】



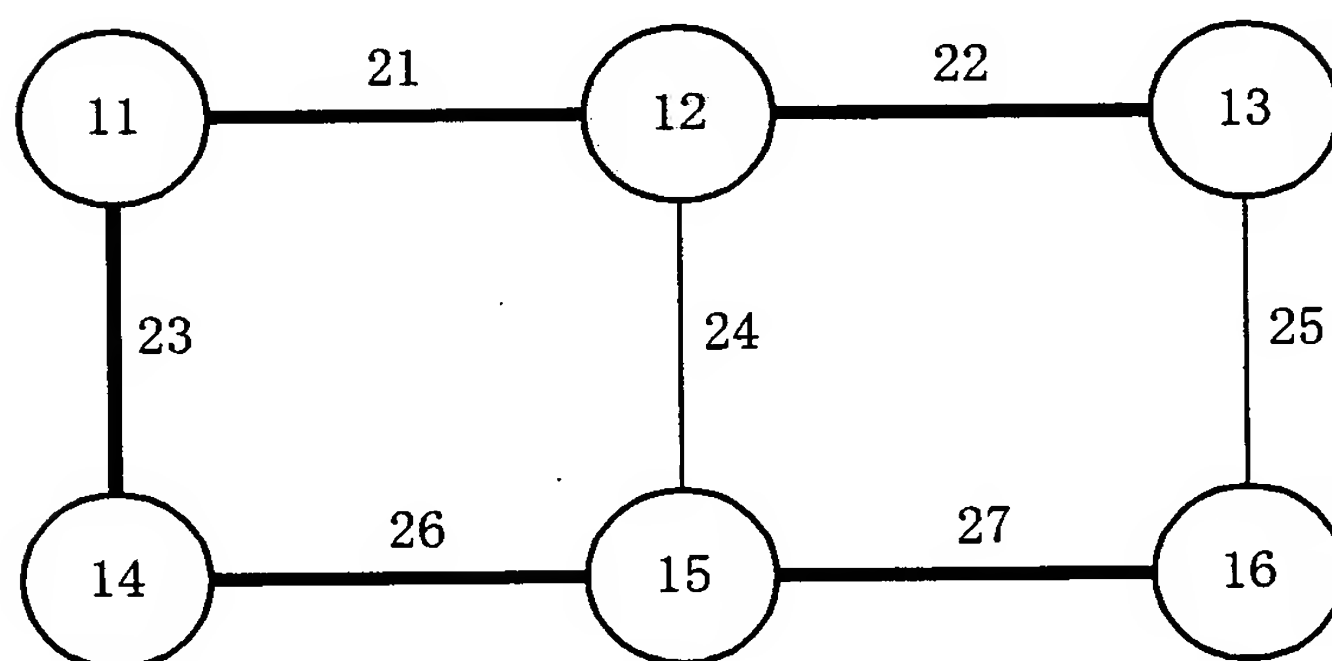
【図 3 7】



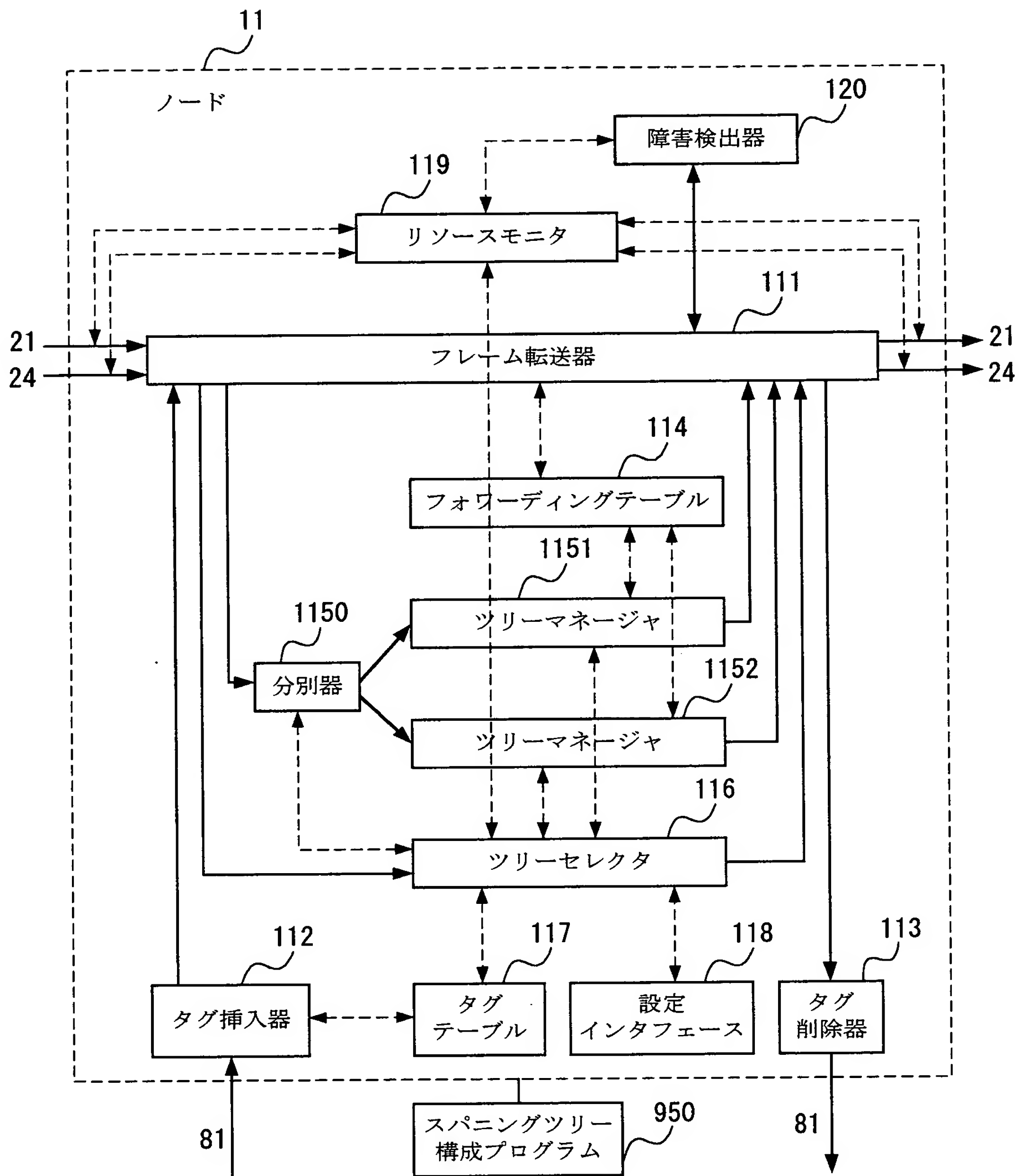
【図 3 8】



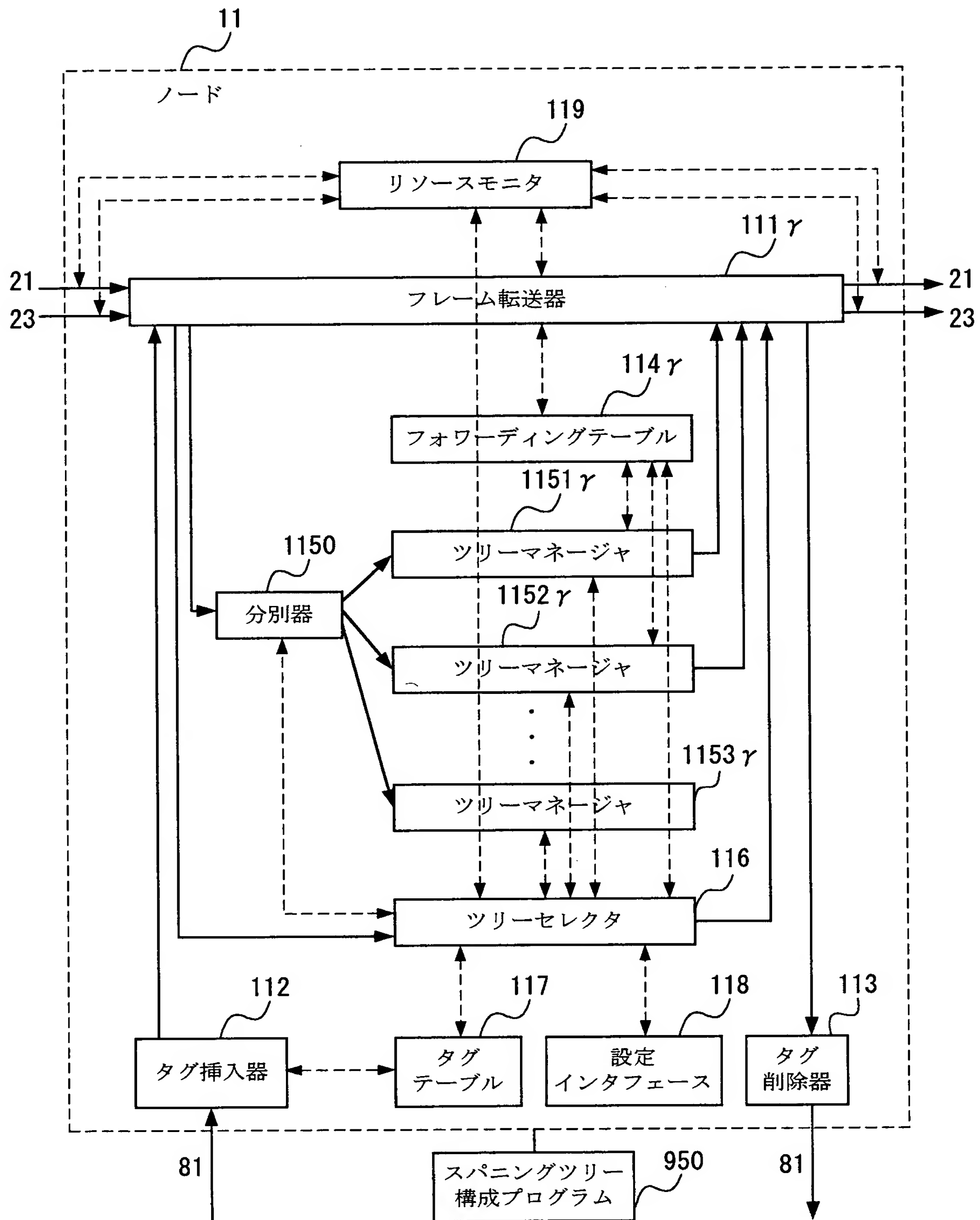
【図 3 9】



【図 4 0】



【図 4 1】

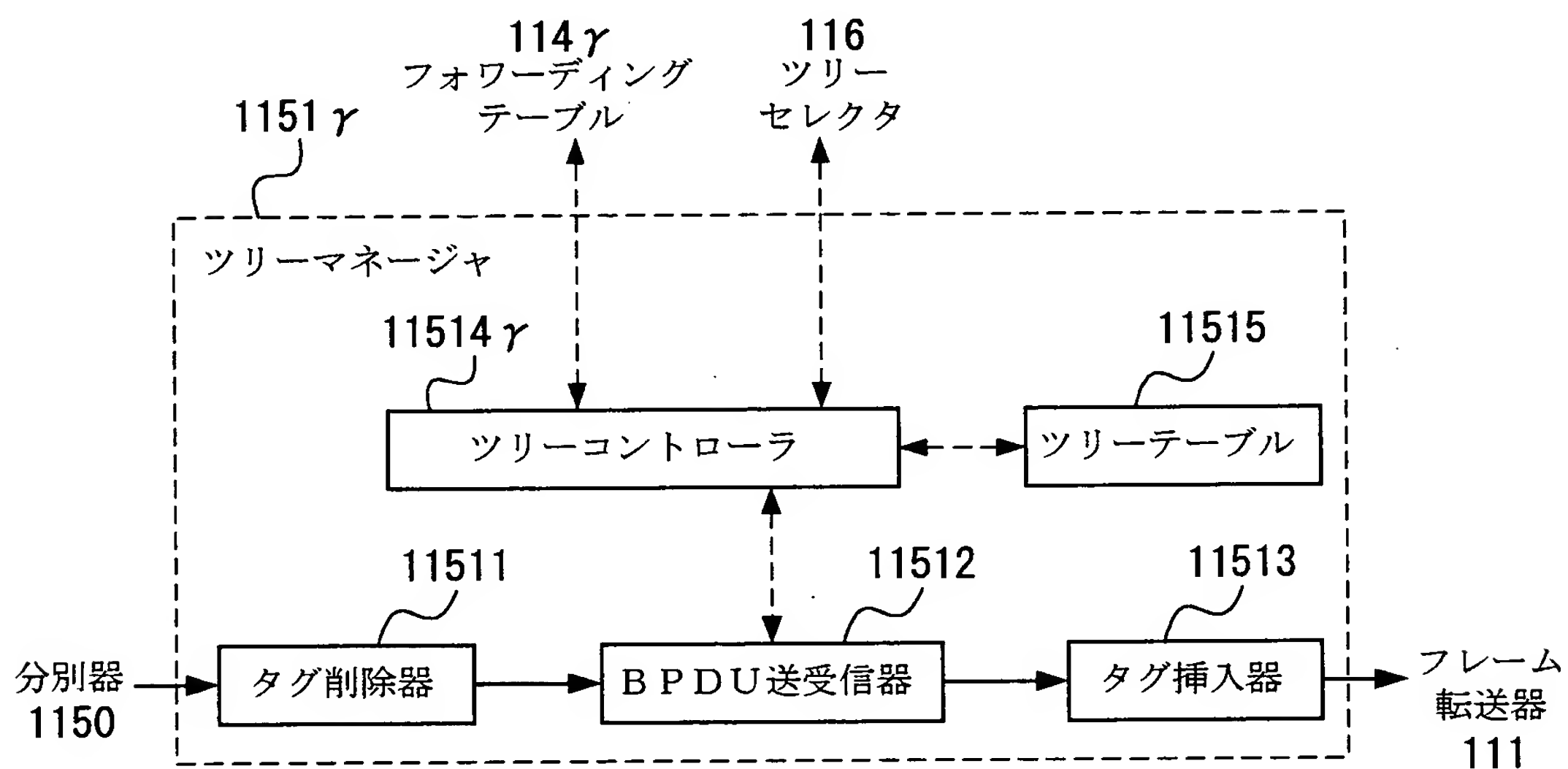


【図 4 2】

114γ フォワーディングテーブル  
(ノード 1 1 から見たビュー)

タグ 1 1 4 1	出力ポート 1 1 4 2
4 1	END
4 2	リンク 2 1 側
4 3	リンク 2 1 側
4 4	リンク 2 3 側
4 5	リンク 2 3 側
4 6	リンク 2 1 側

【図 4 3】





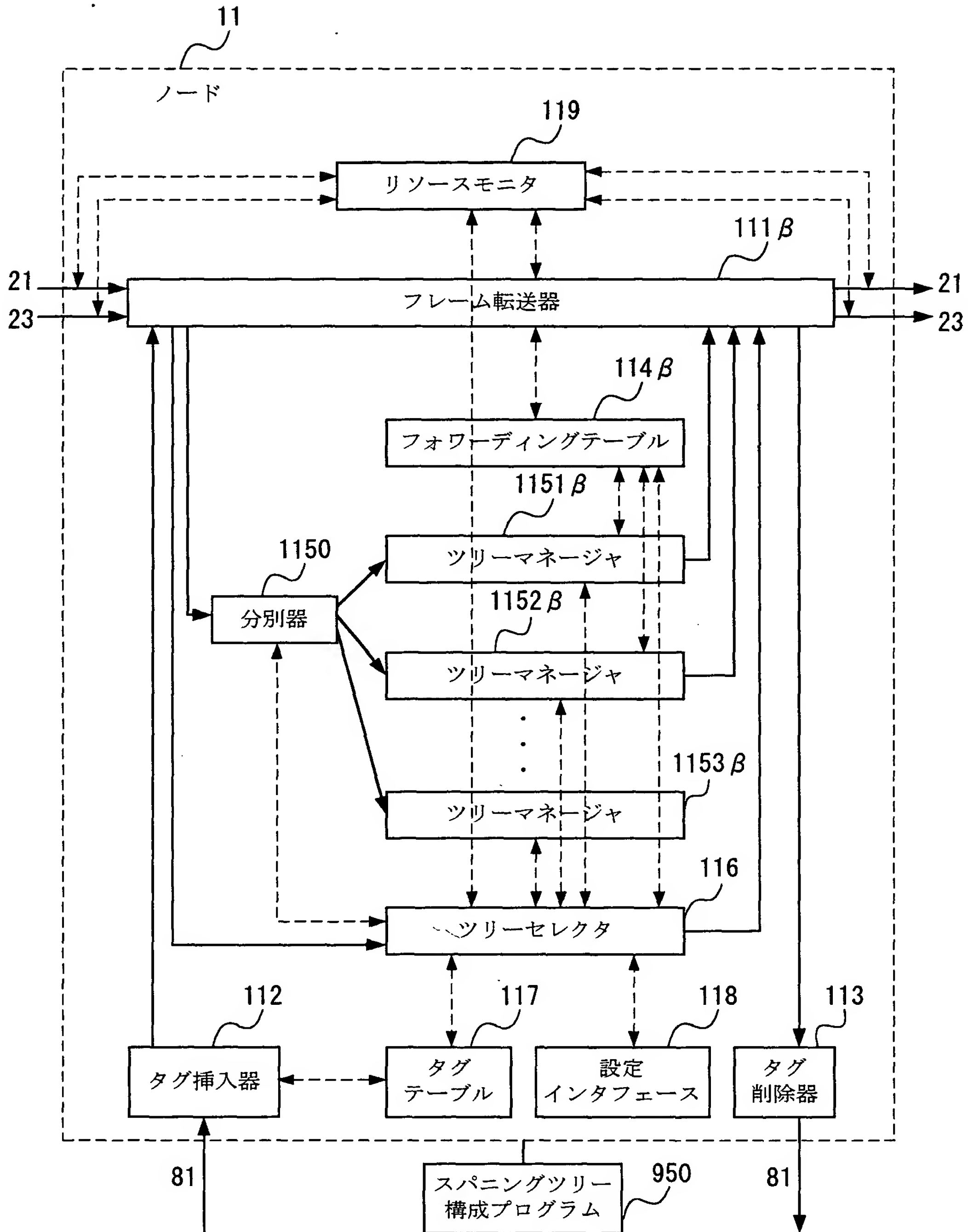
【図 4 4】

スパニングツリー 6 1 (タググループ 4 1) の設定状況



ノード 6 1 0 1	ルートポート 6 1 0 2	タグ 1 1 4 1	出力ポート 1 1 4 2
1 1	なし	4 1	END
1 2	リンク 2 1 側	4 1	リンク 2 1 側
1 3	リンク 2 2 側	4 1	リンク 2 2 側
1 4	リンク 2 3 側	4 1	リンク 2 3 側
1 5	リンク 2 6 側	4 1	リンク 2 6 側
1 6	リンク 2 7 側	4 1	リンク 2 7 側

【図 4 5】



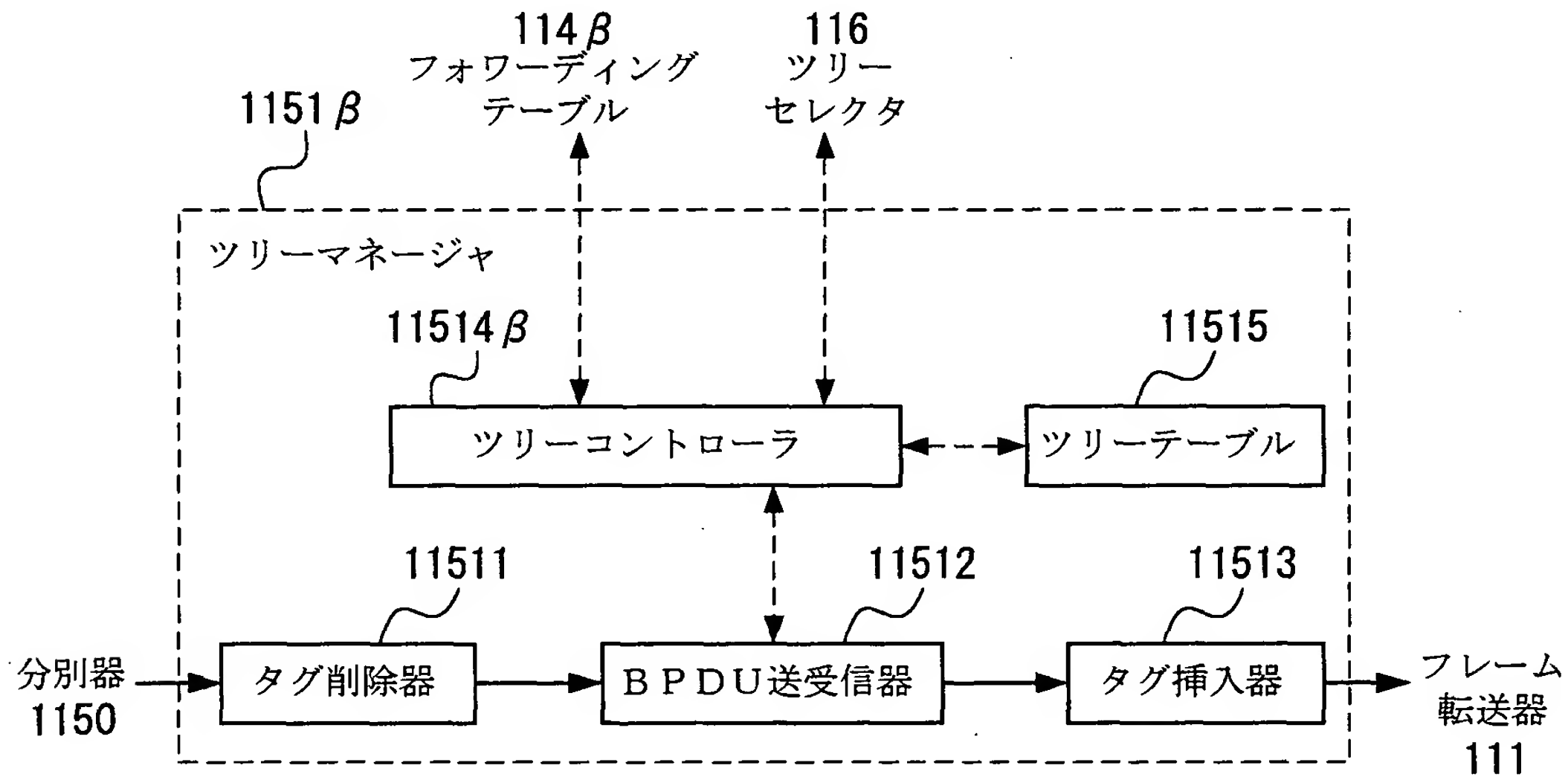
【図 4 6】

114β フォワーディングテーブル

(ノード 1 1 から見たビュー)

タグ 1 1 4 1	出力ポート 1 1 4 2	ブロードキャスト出力ポート 1 1 4 4
4 1	タグ削除器側 (END)	リンク 2 1 側、リンク 2 3 側 (END)
4 2	リンク 2 1 側	リンク 2 3 側 (END)
4 3	リンク 2 1 側	— (END)
4 4	リンク 2 3 側	リンク 2 1 側 (END)
4 5	リンク 2 3 側	— (END)
4 6	リンク 2 1 側	— (END)

【図 4 7】



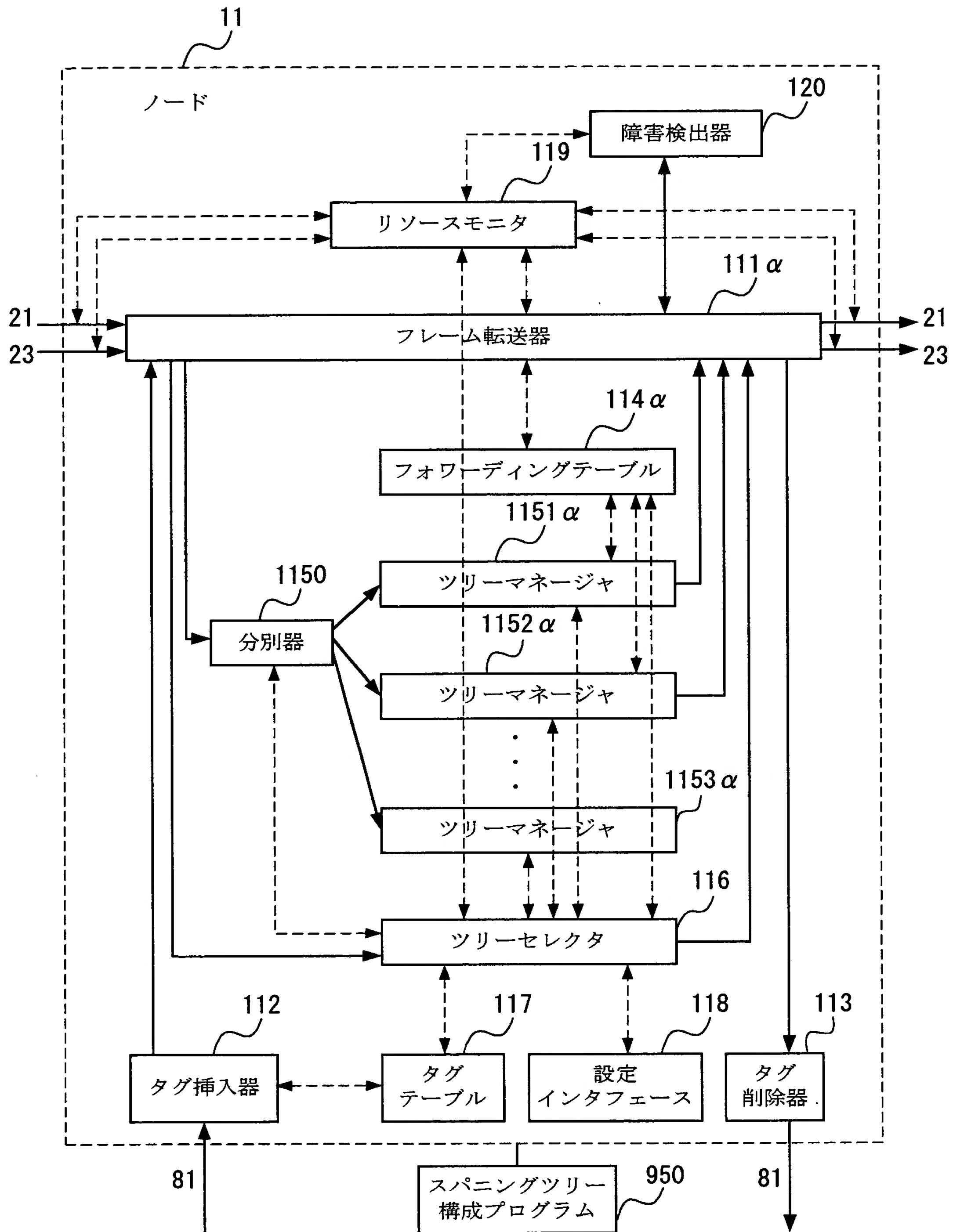
【図 4 8】

スパニングツリー 6 1 (タググループ 4 1) の設定状況



ノード 6 1 0 1	ルートポート 6 1 0 2	ディジグネイ テッドポート 6 1 0 4	タグ 1 1 4 1	出力ポート 1 1 4 2	ブロードキャスト 出力ポート 1 1 4 4
1 1	なし	リンク 2 1 側、 リンク 2 3 側	4 1	タグ削除器側	リンク 2 1 側、 リンク 2 3 側
1 2	リンク 2 1 側	リンク 2 2 側	4 1	リンク 2 1 側	リンク 2 2 側
1 3	リンク 2 2 側	なし	4 1	リンク 2 2 側	—
1 4	リンク 2 3 側	リンク 2 6 側	4 1	リンク 2 3 側	リンク 2 6 側
1 5	リンク 2 6 側	リンク 2 7 側	4 1	リンク 2 6 側	リンク 2 7 側
1 6	リンク 2 7 側	なし	4 1	リンク 2 7 側	—

【図 4 9】

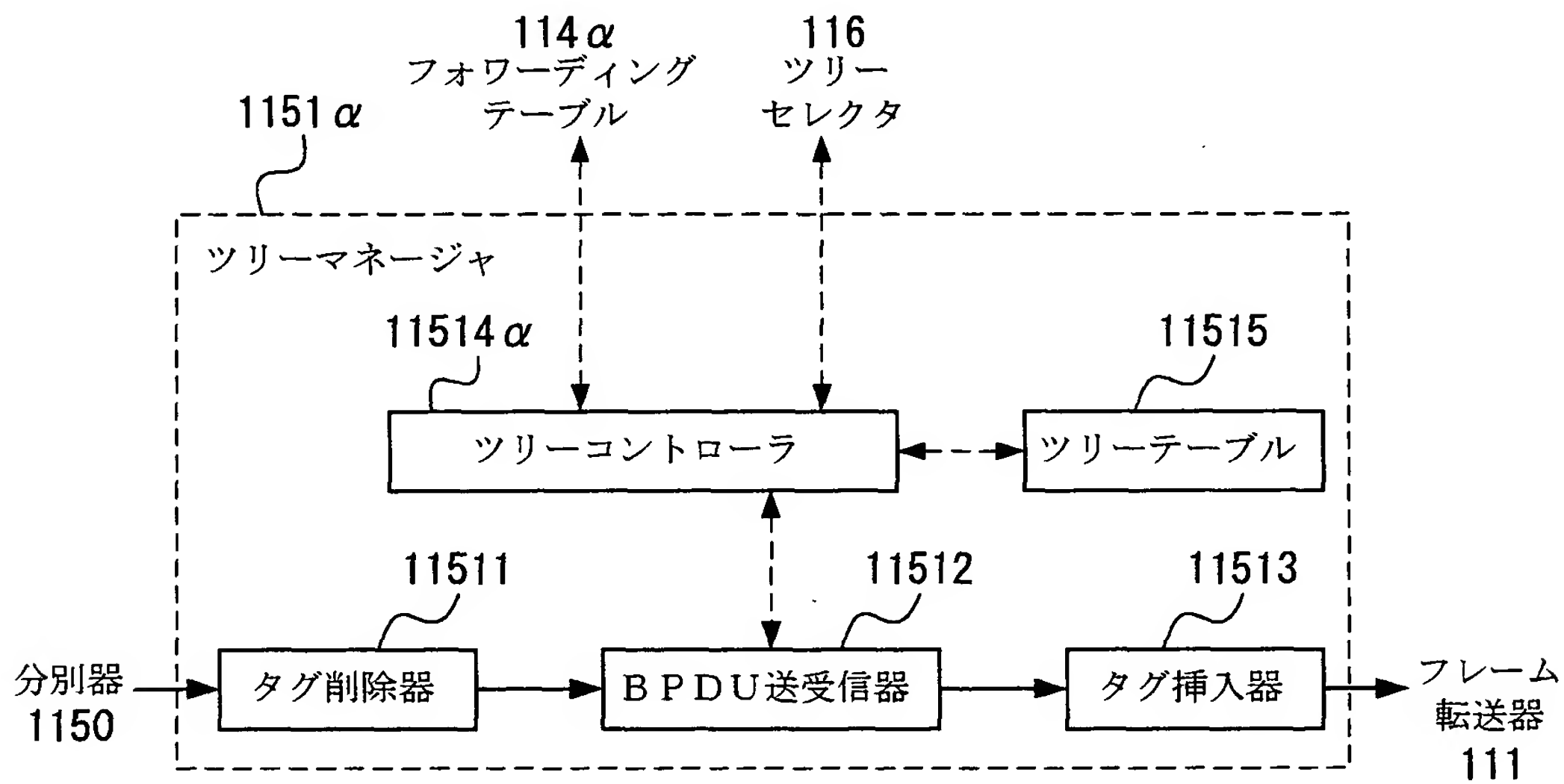


【図 5 0】

114 $\alpha$  フォワーディングテーブル  
(ノード 1 2 から見たビュー)

タグ 1 1 4 1	ルートポート 1 1 4 2	予備出力ポート 1 1 4 3
4 1	リンク 2 1 側	リンク 2 4 側
4 2	END	END
4 3	リンク 2 2 側	リンク 2 4 側
4 4	リンク 2 1 側	リンク 2 4 側
4 5	リンク 2 4 側	リンク 2 1 側 (又は 2 2)
4 6	リンク 2 2 側	リンク 2 4 側

【図 5 1】



【図 5 2】

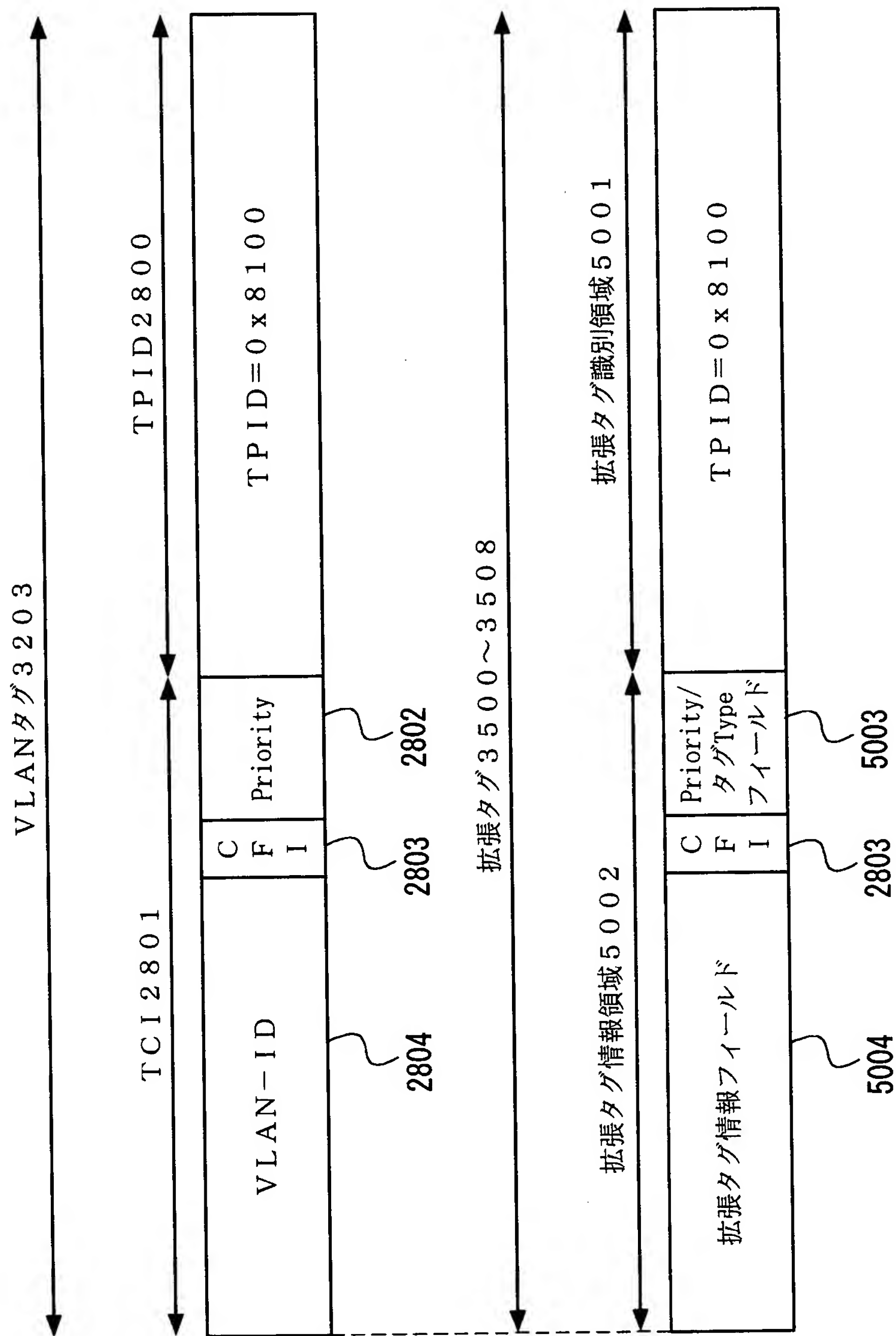
スパニングツリー 6 1 (タググループ 4 1) の設定状況



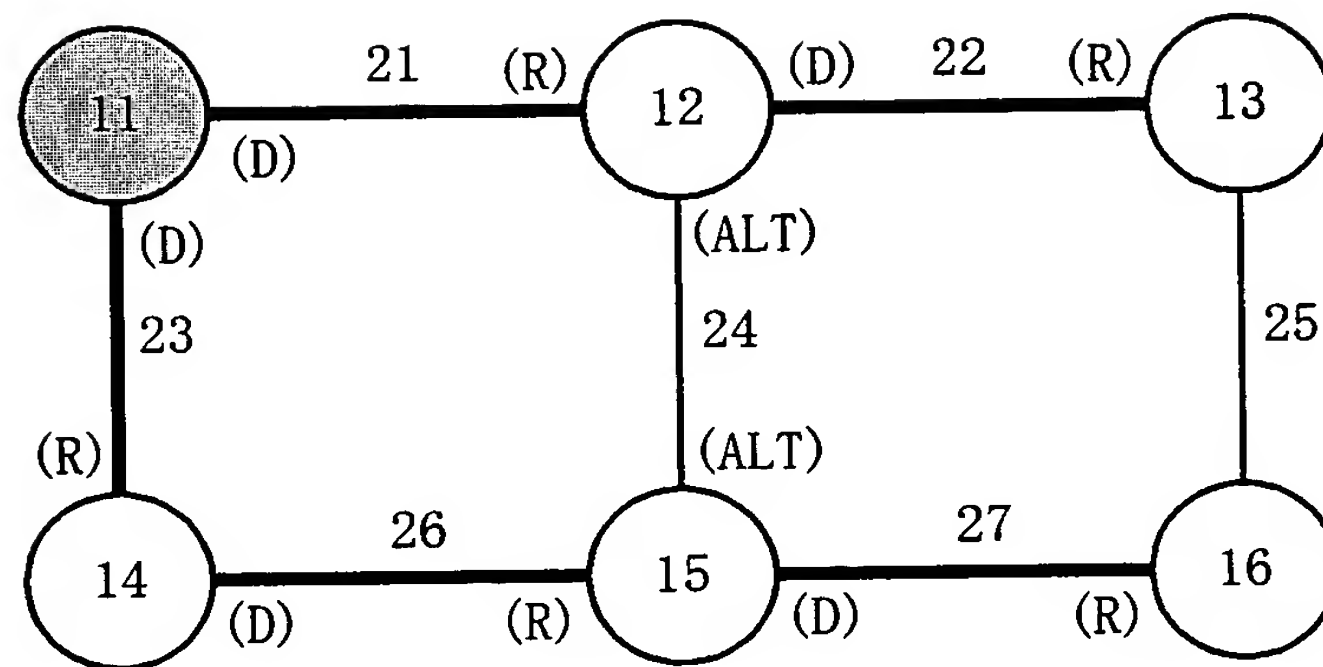
ノード 6 1 0 1	ルートポート 6 1 0 2	オルタネート ポート 6 1 0 3	タグ 1 1 4 1	出力ポート 1 1 4 2	予備出力ポート 1 1 4 3
1 1	なし	なし	4 1	タグ削除器側	—
1 2	リンク 2 1 側	リンク 2 4 側	4 1	リンク 2 1 側	リンク 2 4 側
1 3	リンク 2 2 側	なし	4 1	リンク 2 2 側	—
1 4	リンク 2 3 側	なし	4 1	リンク 2 3 側	—
1 5	リンク 2 6 側	リンク 2 4 側	4 1	リンク 2 6 側	リンク 2 4 側
1 6	リンク 2 7 側	なし	4 1	リンク 2 7 側	—



【図 5 3】



【図 5 4】



【書類名】 要約書

【要約】

【課題】 輻輳発生確率を下げ、輻輳によるフレームの到着遅れや欠落が発生する頻度を減らすことができるネットワークシステム、スパニングツリー構成方法を提案する。

【解決手段】 複数のノードを接続したネットワーク上でスパニングツリーを構成するノードが、変更前のツリーを運用したまま、別のVLANを用いてコスト変更後のツリーを生成し、新規ツリーの安定後に転送に利用するツリーを切り替える。

【選択図】 図7

認 定 ・ 付 加 情 報

特許出願の番号	特願 2 0 0 3 - 0 4 1 8 3 8
受付番号	5 0 3 0 0 2 6 6 9 7 8
書類名	特許願
担当官	第八担当上席 0 0 9 7
作成日	平成 1 5 年 5 月 2 日

< 認定情報・付加情報 >  
【提出日】

平成15年 2月19日

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日 1990年 8月29日  
[変更理由] 新規登録  
住 所 東京都港区芝五丁目7番1号  
氏 名 日本電気株式会社